

Bewährte Verfahrensweisen für den Entwurf effizienter Tableau-Arbeitsmappen

Aktualisiert für Tableau 10

Alan Eldridge
Tableau Software

Info zu diesem Dokument

Ich möchte nochmals darauf hinweisen, dass dieses Dokument auf Unterlagen von zahlreichen Autoren basiert. Ich habe ihre Arbeiten vor allem in einem Dokument zusammengetragen und strukturiert. Einige Personen werden bei der Lektüre diverser Abschnitte ihre Beiträge wiedererkennen (einige werden sogar weite Teile ihrer Texte wiederfinden). Ihnen allen möchte ich danken, weil dieses Dokument ohne Ihre kontinuierliche, ausgezeichnete Arbeit und ohne den Verzicht auf Urheberrechtsansprüche nicht existieren würde.

Ich möchte auch zahlreichen Personen danken, die dieses Dokument auf Genauigkeit und Lesbarkeit geprüft haben. Ihre Detailgenauigkeit und geduldigen Erläuterungen haben dieses Dokument lesbarer gemacht, als ich es selbst hätte machen können.

Dieses Dokument wurde hinsichtlich der Funktionen von Tableau 10 aktualisiert. Zukünftige Versionen von Tableau werden neue Features und Funktionen bieten, die möglicherweise einige dieser Empfehlungen verändern.

Vielen Dank!

Alan Eldridge

Juni 2016

TL;DR

Von Personen, die dieses Whitepaper gelesen oder empfohlen haben, wurde mir häufig gesagt, dass es zu lang ist. Meine Antwort ist, dass es so lang ist, wie es sein muss, um derart umfangreiches Material mit angemessener Gründlichkeit zu behandeln.

Es folgt eine Auflistung wichtiger Erkenntnisse aus diesem Dokument:

- Es gibt keinen Königsweg für effiziente Arbeitsmappen. Sehen Sie sich zunächst die Leistungsaufzeichnung an, damit Sie verstehen, wo Zeit erforderlich ist. Lang dauernde Abfragen? Viele Abfragen? Langsame Berechnungen? Komplexes Rendern? Verwenden Sie diese Erkenntnis, um Ihre Bemühungen in die richtige Richtung zu lenken.
- Die Empfehlungen in diesem Dokument sind genau das: Empfehlungen. Einerseits repräsentieren sie bewährte Verfahrensweisen, die Sie benötigen, um zu testen, ob sie die Leistung in Ihrem speziellen Fall verbessern. Viele von ihnen können von der Struktur Ihrer Daten und von der von Ihnen verwendeten Datenquelle abhängen (z. B. unstrukturierte Datei vs. RDBMS vs. Datenextrakt).
- Extrakte sind eine schnelle und einfache Möglichkeit, um die meisten Arbeitsmappen zu beschleunigen.
- Je sauberer Ihre Daten sind und je besser sie mit der Struktur Ihrer Fragen übereinstimmen (d. h. je weniger Vorbereitung und Bearbeitung erforderlich ist), desto schneller laufen Ihre Arbeitsmappen.
- Die Ursache für langsame Dashboards ist meistens bei schlechtem Design zu finden: insbesondere zu viele Grafiken in einem Dashboard oder der Versuch, zu viele Daten gleichzeitig anzuzeigen. Keep it simple. Ermöglichen Sie es Ihren Benutzern, stufenweise Detaildaten aufzurufen, anstatt zu versuchen, alles zu zeigen und dann zu filtern.
- Arbeiten Sie mit den Daten, die Sie benötigen, und nicht mehr. Sowohl im Hinblick auf die Felder, auf die Sie sich beziehen, als auch auf die Granularität der zurückgegebenen Datensätze. So kann Tableau weniger, bessere, schnellere Abfragen generieren und den Umfang der Daten reduzieren, die von der Datenquelle in die Tableau-Engine bewegt werden müssen. Damit wird auch die Größe Ihrer Arbeitsmappen reduziert, sodass sie leichter freizugeben sind und sich schneller öffnen.
- Wenn Sie die Daten reduzieren, achten Sie darauf, die Filter effizient zu nutzen.
- Zeichenfolgen und Datumsangaben sind langsam, Zahlen und boolesche Werte sind schnell.

Zu guter Letzt haben einige der Empfehlungen in diesem Dokument nur eine materielle Auswirkung, wenn Sie mit großen und/oder komplexen Datenbeständen arbeiten. Was bedeutet groß oder komplex? Es kommt darauf an. Es schadet jedoch nicht, diese Empfehlungen in allen Ihren Arbeitsmappen zu berücksichtigen, weil Sie nie wissen, wann Ihre Daten anwachsen werden. Übung macht den Meister.

Inhalt

Info zu diesem Dokument	2
TL;DR	3
Einleitung	6
Wofür ist Tableau gut?.....	6
Wofür ist Tableau nicht so gut?	6
Effizienz verstehen	8
Was ist eine „effiziente“ Arbeitsmappe?.....	8
Warum sollten Sie sich Gedanken um Effizienz machen?	8
Die Gesetze der Physik.....	9
Handwerkszeug.....	11
Leistungsaufzeichnung.....	11
Protokolle.....	13
Tableau Server Leistungsansichten.....	14
Überwachen und testen	15
Weitere Tools.....	16
Liegt es am Design meiner Arbeitsmappe?	18
Gutes Dashboard-Design	18
Tunen Sie Ihr Dashboard auf Leistung	22
Gutes Arbeitsblattdesign	27
Effiziente Filter	33
Berechnungstypen	44
Analysen.....	48
Berechnungen im Vergleich zu nativen Funktionen	49
Einfluss von Datentypen	49
Leistungstechniken	49
Liegt es an meinen Abfragen?.....	55
Automatische Optimierungen	55
Verknüpfungen	62
Verschmelzung.....	62
Datenintegration.....	66
SQL (benutzerdefiniert)	68
Alternativen zu benutzerdefinierter SQL.....	69
Liegt es an meinen Daten?.....	71
Allgemeiner Ratschlag.....	71
Datenquellen.....	72

Datenvorbereitung.....	81
Datenextrakte	82
Data Governance	89
Liegt es an meiner Umgebung?	90
Upgrade durchführen	90
Testen Sie Tableau Desktop auf dem Server	90
Separate Aktualisierungen und interaktive Arbeitslasten	90
Tableau Server überwachen und feinabstimmen.....	91
Infrastruktur.....	91
Fazit.....	93

Einleitung

Wofür ist Tableau gut?

Bei Tableau versuchen wir zu verändern, wie die Menschen Daten sehen, wie sie mit ihnen interagieren und sie verstehen. Deshalb versuchen wir nicht, die gleiche Erfahrung zu liefern wie traditionelle BI-Unternehmensplattformen. Tableau kann seine Stärken voll ausspielen, wenn Sie Arbeitsmappen erstellen, die die folgenden Merkmale aufweisen:

- **Visuell** – eine große Zahl von Untersuchungen belegen, dass der Mensch große, komplexe Datenvolumen auf effizienteste Weise über eine visuelle Darstellung versteht. Tableau präsentiert Daten vor allem über Diagramme, Grafiken und Dashboards. Tabellen und Kreuztabellen haben ihren eigenen Platz (und werden unterstützt). Wir werden uns später damit befassen, wie sie am besten einzusetzen sind.
- **Interaktiv** – Tableau-Dokumente sind für interaktive Präsentation konzipiert, entweder auf dem Desktop, über das Web oder über ein Mobilgerät. Andere BI-Tools produzieren vor allem gedruckte Dokumente (entweder tatsächlich auf Papier oder in Form einer PDF-Datei). Bei Tableau liegt der Fokus auf der Gestaltung reichhaltiger, interaktiver Erfahrungen, mit deren Hilfe der Benutzer Daten untersuchen kann und außerdem durch geschäftsbezogene Fragestellungen begleitet wird.
- **Iterativ** – Das Entdecken ist ein von Natur aus zyklischer Prozess. Tableau ist so konzipiert, dass der Zyklus Frage-Erkenntnis-Frage beschleunigt wird, sodass die Benutzer schnell eine Hypothese entwickeln, sie mit verfügbaren Daten testen, die Hypothese überarbeiten, sie erneut testen können usw.
- **Schnell** – BI-Prozesse waren bisher langsam. Langsam beim Installieren und Konfigurieren der Software, langsam beim Bereitstellen der Daten für die Analyse und langsam beim Konzipieren und Implementieren von Dokumenten, Berichten, Dashboards usw. Mit Tableau können die Benutzer ihre Dokumente schneller als je zuvor erstellen. In vielen Fällen wird die Produktion einer Antwort von Monaten oder Wochen auf Stunden oder Minute verkürzt.
- **Einfach** – Traditionelle BI-Tools für Unternehmen übersteigen häufig die Kompetenzen der meisten Geschäftsanwender, entweder aufgrund der Kosten oder aufgrund der Komplexität. In vielen Fällen benötigen die Benutzer die Unterstützung der IT-Abteilung oder eines Power-Benutzers, der ihnen beim Erstellen der Abfragen und gewünschten Dokumente hilft. Tableau bietet eine intuitive Schnittstelle für nicht technisch versierte Benutzer, um komplexe Daten abzufragen und zu analysieren, ohne Experte für Datenbanken oder Tabellenkalkulation werden zu müssen.
- **Schön** – Die Schönheit liegt zwar bekanntlich im Auge des Betrachters, aber bei der visuellen Kommunikation gibt es bewährte Verfahrensweisen, die befolgt werden sollten. Durch Funktionen wie „Zeig es mir!“ begleitet Tableau Benutzer beim Erstellen effizienter, verständlicher Diagramme auf der Grundlage der verwendeten Daten.
- **Allgegenwärtig** – Die Benutzer erstellen Dokumente nicht mehr nur für eine Ausgabeplattform. Sie müssen Daten auf unterschiedliche Weise anzeigen und mit ihnen interagieren können: auf ihren Desktops, über das Web, auf mobilen Geräten, eingebettet in andere Anwendungen und Dokumente usw. Mit Tableau kann ein einzelnes Dokument veröffentlicht und dann ohne Portierung oder Umformatierung auf all diesen Plattformen benutzt werden.

Wofür ist Tableau nicht so gut?

Tableau ist ein reichhaltiges und leistungsstarkes Tool, man muss jedoch von vornherein verstehen, dass es für bestimmte Probleme möglicherweise nicht die beste Lösung ist. Das bedeutet nicht,

dass es diese Dinge nicht bewerkstelligen kann: Tableau kann für viele Aufgaben eingesetzt werden, für die es ursprünglich nicht konzipiert wurde. Es bedeutet lediglich, dass Tableau nicht für die Lösung dieser Art von Problemen entwickelt wurde, dass möglicherweise das Verhältnis von Aufwand zu Ergebnis ungünstig ausfällt und die resultierende Lösung eventuell nicht gut funktioniert oder nicht flexibel ist.

Überprüfen Sie Ihre Anforderungen, oder ziehen Sie einen anderen Ansatz in Betracht, falls Folgendes zutrifft:

- Sie benötigen ein Dokument, das für die Papiausgabe, nicht für den Bildschirm konzipiert wurde. Sie müssen beispielsweise ein komplexes Seitenlayout steuern, mit Funktionen wie Kopfzeilen/Fußzeilen für Seiten, Abschnitte und Gruppen oder präzise WYSIWYG-Formatierung. Tableau kann mehrseitige Berichte produzieren, bietet allerdings nicht die Formatierungskontrolle, die in spezifischen Tools für fortlaufende Berichte verfügbar sind.
- Sie benötigen einen komplexen Push-Mechanismus für personalisierte Dokumente (sogenanntes „Bursting“), die über verschiedene Bereitstellungsmodi versendet werden. Tableau Server beinhaltet das Konzept der Berichtsabonnements, mit denen ein Benutzer Berichte abonnieren kann (und in Tableau 10 auch Abonnements für andere Benutzer einrichten), die er dann per E-Mail bekommt. Die Kunden wünschen sich jedoch häufig eine flexiblere Lösung. Tableau kann für andere Formen von Push-Delivery-Systemen eingesetzt werden, jedoch ist dies keine eigentliche Funktion von Tableau. Dies erfordert die Entwicklung einer benutzerdefinierten Lösung auf der Grundlage des Dienstprogramms TABCMD oder die Einführung externer Lösungen wie etwa *VizAlerts* (<http://tabsoft.co/1stldFh>) oder *Push Intelligence for Tableau* von Metric Insights (<http://bit.ly/1HACxul>).
- Beim Reader geht es primär darum, Daten in ein anderes Format zu exportieren (häufig eine CSV- oder Excel-Datei). Dies bedeutet häufig einen Bericht in Tabellenform mit vielen Zeilen und detaillierten Daten. Zur Klarstellung: Bei Tableau können die Benutzer Daten aus einer Ansicht oder einem Dashboard nach Excel exportieren, entweder auf der Zusammenfassungs- oder der Detailebene. Wenn es allerdings primär um das Exportieren von Daten geht, ist dies nur ein zweitrangiger Ersatz für den ETL-Prozess (Extrahieren, Transformieren, Laden). Es gibt viel effizientere Lösungen als ein Berichtstool, um dies zu erreichen.
- Sie benötigen hochkomplexe Dokumente in Kreuztabellenform, die möglicherweise vorhandene Tabellenkalkulationsberichte mit komplexen Zwischensummen und Verweisen usw. widerspiegeln. Häufig auftretende Beispiele sind Finanzberichte wie GuV, Bilanz usw. Außerdem kann eine Szenario-Modellierung, eine Was-wäre-wenn-Analyse und sogar das Zurückschreiben von Annahmedaten erforderlich sein. Wenn die zugrunde liegenden granularen Daten nicht verfügbar sind oder wenn die Berichtslogik auf Zellenverweisen basiert und nicht auf der Zusammenfassung von Datensätzen in Gesamtsummen, kann es besser sein, für diese Berichtsart weiterhin eine Tabellenkalkulation zu benutzen.

Effizienz verstehen

Was ist eine „effiziente“ Arbeitsmappe?

Mehrere Faktoren machen eine Arbeitsmappe „effizient“. Einige dieser Faktoren sind technik- und andere eher benutzerorientiert, aber im Allgemeinen zeichnet sich eine effiziente Arbeitsmappe durch folgende Merkmale aus:

- **Einfach** – Ist es einfach, die Arbeitsmappe zu erstellen und wird sie in Zukunft einfach zu pflegen sein? Nutzt sie die Prinzipien der visuellen Analyse, um die Botschaft des Autors und der Daten deutlich zu vermitteln?
- **Flexibel** – Kann die Arbeitsmappe mehrere Fragen des Benutzers beantworten oder nur eine? Taucht der Benutzer in eine interaktive Erfahrung ein oder ist es nur ein statischer Bericht?
- **Schnell** – Reagiert die Arbeitsmappe schnell genug für die Benutzer? Dies bezieht sich auf die Zeit zum Öffnen, zum Aktualisieren oder zum Reagieren auf eine Interaktion. Dies kann ein etwas subjektiver Blickwinkel sein, aber im Allgemeinen soll die Arbeitsmappe innerhalb weniger Sekunden eine erste Anzeige von Daten bieten und auf Benutzereingaben reagieren.

Die Leistung eines Dashboards wird von folgenden Aspekten beeinflusst:

- Das visuelle Design auf der Dashboard- und Arbeitsmappen Ebene – d. h. wie viele Elemente, wie viele Datenpunkte, Nutzung von Filtern und Aktionen usw.
- Die Berechnungen – z. B. welche Art von Berechnung, wo wird die Berechnung ausgeführt usw.
- Die Abfragen – z. B. wie viele Daten werden zurückgegeben, handelt es sich um SQL (benutzerdefiniert)
- Die Datenverbindungen und zugrunde liegenden Datenquellen
- Einige Unterschiede zwischen Tableau Desktop und Tableau Server
- Sonstige Umgebungsfaktoren, wie etwa Hardwarekonfiguration und Kapazität

Warum sollten Sie sich Gedanken um Effizienz machen?

Sie sollten aus mehreren Gründen darüber nachdenken:

- Wenn Sie als Analyst oder Arbeitsmappenautor effizient arbeiten, gelangen Sie schneller zur gewünschten Antwort.
- Effizient zu arbeiten bedeutet, dass Sie im „Flow“ der Analyse bleiben. Das bedeutet, dass Sie über Fragen und Daten nachdenken, anstatt darüber, wie Sie mit dem Tool zu einem Ergebnis kommen.
- Eine Arbeitsmappe mit einem flexiblen Design zu erstellen, reduziert die Notwendigkeit, mehrere Arbeitsmappen zu erstellen und zu pflegen, die ähnliche Anforderungen haben.
- Eine Arbeitsmappe mit einem einfachen Design zu erstellen, erleichtert es anderen, Ihre Arbeitsmappe zu übernehmen und weitere Iterationen Ihrer Arbeit zu entwickeln.
- Der Eindruck von Reaktionsfähigkeit ist ein wichtiger Erfolgsfaktor für Endbenutzer, wenn sie sich Berichte und Dashboards ansehen. Wenn Ihre Arbeitsmappen also so schnell wie möglich laufen, sorgen Sie für zufriedene Benutzer.

Unsere umfassende Erfahrung hat gezeigt, dass die meisten Leistungsprobleme bei Kunden auf Designfehler in den Arbeitsmappen beruhen. Wenn wir diese Fehler beheben oder durch Schulung von vornherein vermeiden können, können wir die Probleme beheben.

Wenn Sie mit kleinen Datenvolumen arbeiten, sind viele dieser Empfehlungen nicht kritisch. Sie können das Problem einfach mit der Holzhammermethode lösen. Wenn Sie es jedoch mit hunderten von Millionen Datensätzen, mehreren Arbeitsmappen oder vielen Autoren zu tun haben, werden die

Auswirkungen eines schlechten Arbeitsmappendesigns verstärkt. Dann müssen Sie die Hinweise in diesem Whitepaper stärker berücksichtigen.

Natürlich macht Übung den Meister und es wird empfohlen, diese Anleitungen bei allen Arbeitsmappen zu befolgen. Nicht vergessen: Ihr Design ist erst abgeschlossen, nachdem Sie Ihre Arbeitsmappe mit den erwarteten Produktionsdatenvolumen getestet haben.

Wichtiger Hinweis: Wir beziehen uns in diesem Dokument auf Tableau Server, aber in den meisten Fällen gelten die Anweisungen auch für Tableau online, falls Sie unsere gehostete Lösung einer standortgebundenen Bereitstellung vorziehen. Die offensichtlichen Ausnahmen sind die Hinweise zur Feinabstimmung der Serverkonfiguration und zum Installieren/Aktualisieren der Software auf der Serverebene (in der SaaS-Welt brauchen Sie sich darum nicht zu kümmern).

Die Gesetze der Physik

Bevor wir uns in technische Details vertiefen, um aufzuzeigen, wie die verschiedenen Funktionen die Leistung von Arbeitsmappen beeinflussen, gibt es einige Grundsätze, die Ihnen beim Entwerfen effizienter Dashboards und Berichte helfen werden:

Wenn die Leistung in der Datenquelle schwach ist, wird sie in Tableau nicht besser sein.

Wenn Ihre Tableau-Arbeitsmappe auf einer langsamen Abfrage beruht, wird Ihre Arbeitsmappe ebenfalls langsam sein. In den folgenden Abschnitten finden Sie Tuning-Tipps für Ihre Datenbanken, damit Abfragen schneller ablaufen. Außerdem wird erörtert, wie die schnelle Daten-Engine von Tableau eingesetzt werden kann, um die Abfrageleistung zu verbessern.

Wenn die Leistung in Tableau Desktop schwach ist, wird sie (fast immer) in Tableau Server nicht besser sein.

Eine Arbeitsmappe, die in Tableau Desktop langsam abläuft, wird durch die Veröffentlichung in Tableau Server nicht schneller. Benutzer haben häufig den Eindruck, dass Ihre Arbeitsmappe in Tableau Server schneller laufen wird, weil der Server mehr CPU/RAM/usw. als ihr lokaler PC hat. Im Allgemeinen laufen die Arbeitsmappen auf Tableau Server aus folgenden Gründen etwas langsamer:

- Mehrere Benutzer nutzen die Server-Ressourcen gleichzeitig, um Arbeitsmappen zu generieren (obwohl Sie intuitiv möglicherweise den Eindruck haben, dass Ihre Arbeitsmappe schneller reagiert, wenn Sie sie freigegeben haben. Der Grund dafür sind die Caching-Mechanismen in Tableau Server)
- Der Server muss zusätzliche Arbeit leisten, um die Dashboards und Grafiken zu rendern, weil dies nicht auf der Client-Workstation erfolgt.

Sie sollten zuerst versuchen, die Feinabstimmung Ihrer Arbeitsmappe in Tableau Desktop vorzunehmen, bevor Sie versuchen, die Leistung in Tableau Server zu verbessern.

Die Ausnahme für diese Regel ist der Fall, dass Tableau Desktop begrenzte Ressourcen zur Verfügung hat, was auf Tableau Server nicht der Fall ist. Etwa falls Ihr PC nicht über genügend RAM verfügt, um das Datenvolumen zu bewältigen, das Sie analysieren, oder der Server eine schnellere Verbindung zur Datenquelle mit weniger Latenzen hat. Bei einigen Benutzern treten geringe Leistung oder sogar die Fehlermeldung „Nicht genügend Arbeitsspeicher“ auf, wenn sie auf ihrer Workstation mit einem Datensatz und nur 2 GB RAM arbeiten, jedoch feststellen, dass die veröffentlichte Arbeitsmappe mit akzeptabler Geschwindigkeit funktioniert, weil der Server über viel mehr Speicher und Verarbeitungsleistung verfügt.

Neuer ist besser

Das Entwicklungsteam bei Tableau befasst sich ständig mit der Verbesserung der Leistung und Benutzerfreundlichkeit unserer Software. Ein Upgrade auf die neueste Version von Tableau Desktop und Tableau Server kann gelegentlich signifikante Leistungsverbesserungen zur Folge haben, ohne dass die Arbeitsmappe geändert werden muss. Viele Kunden berichteten z. B. von dreifachen (und mehr) Leistungsverbesserungen bei ihren Arbeitsmappen durch ein Upgrade von V8 auf V9 und die Leistungsverbesserung steht auch bei Tableau 10 im Fokus. Bei Tableau Online ist dies natürlich nicht von Bedeutung, weil es immer auf die neueste Version aktualisiert wird, sobald sie veröffentlicht wird.

Dies gilt für Wartungsversionen sowie für große und kleine Updates. Weitere Informationen über den Versionszyklus von Tableau sowie spezifische Details zu den einzelnen Versionen finden Sie auf dieser Seite mit Versionshinweisen:

<http://www.tableau.com/de-de/support/releases>

Außerdem arbeiten Datenbankanbieter daran, ihre Angebot zu verbessern. Achten Sie darauf, dass Sie auch die neuesten Versionen der geeigneten Datenquellentreiber verwenden. Sie finden sie auf der folgenden Webseite:

<http://www.tableau.com/de-de/support/drivers>

Weniger ist mehr

Hier gilt die gleiche Regel wie allgemein im Leben: zu viel des Guten kann schlecht sein.

Versuchen Sie nicht, absolut alles in einer einzigen, monolithischen Arbeitsmappe unterzubringen. Eine Tableau-Arbeitsmappe kann zwar 50 Dashboards umfassen, die jeweils wieder 20 Diagrammobjekte haben und mit 50 verschiedenen Datenquellen verbunden sind. Aber Sie können sicher sein, dass sie langsam sein wird.

Wenn Sie eine solche Arbeitsmappe haben, sollten Sie sie in mehrere separate Dateien aufteilen. Das ist ganz einfach: Sie können Dashboards einfach zwischen Arbeitsmappen kopieren und Tableau fügt alle dazugehörigen Arbeitsblätter und Datenquellen hinzu. Wenn Ihre Dashboards zu komplex sind, vereinfachen Sie sie mithilfe von Interaktionen, die den Benutzer von Bericht zu Bericht führen. Vergessen Sie nicht, dass der Preis für unsere Software nicht von der Anzahl der Dokumente abhängig ist, also breiten Sie die Daten ruhig ein wenig aus.

Skalierbarkeit ist nicht das Gleiche wie Leistung

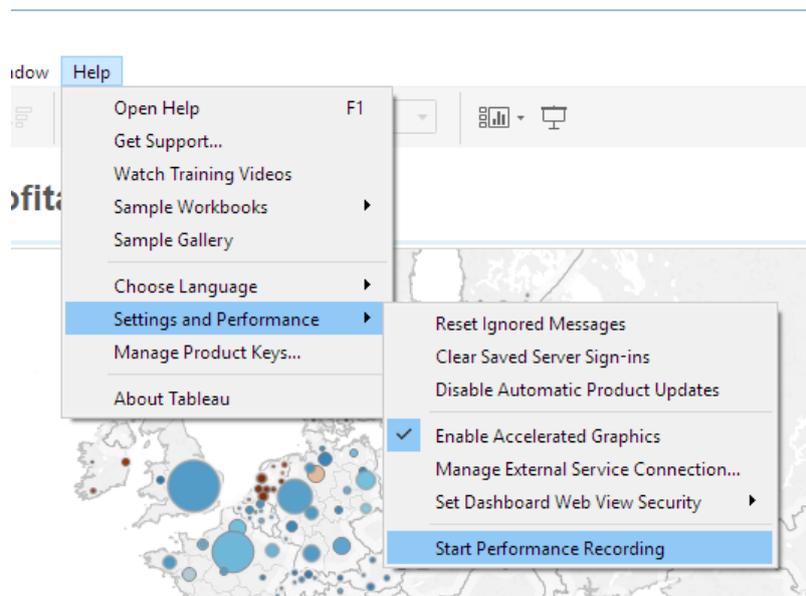
Bei Skalierbarkeit geht es darum, dass wir mehrere Benutzer unterstützen können, die freigegebene Arbeitsmappen ansehen. Bei Leistung geht es darum, dass eine einzelne Arbeitsmappe so schnell wie möglich funktioniert. Während die meisten angebotenen Empfehlungen in diesem Dokument eine positive Auswirkung auf die Skalierbarkeit für Arbeitsmappen haben, die auf Tableau Server veröffentlicht wurden, liegt der Hauptfokus dieses Dokuments auf Leistungsverbesserung.

Handwerkszeug

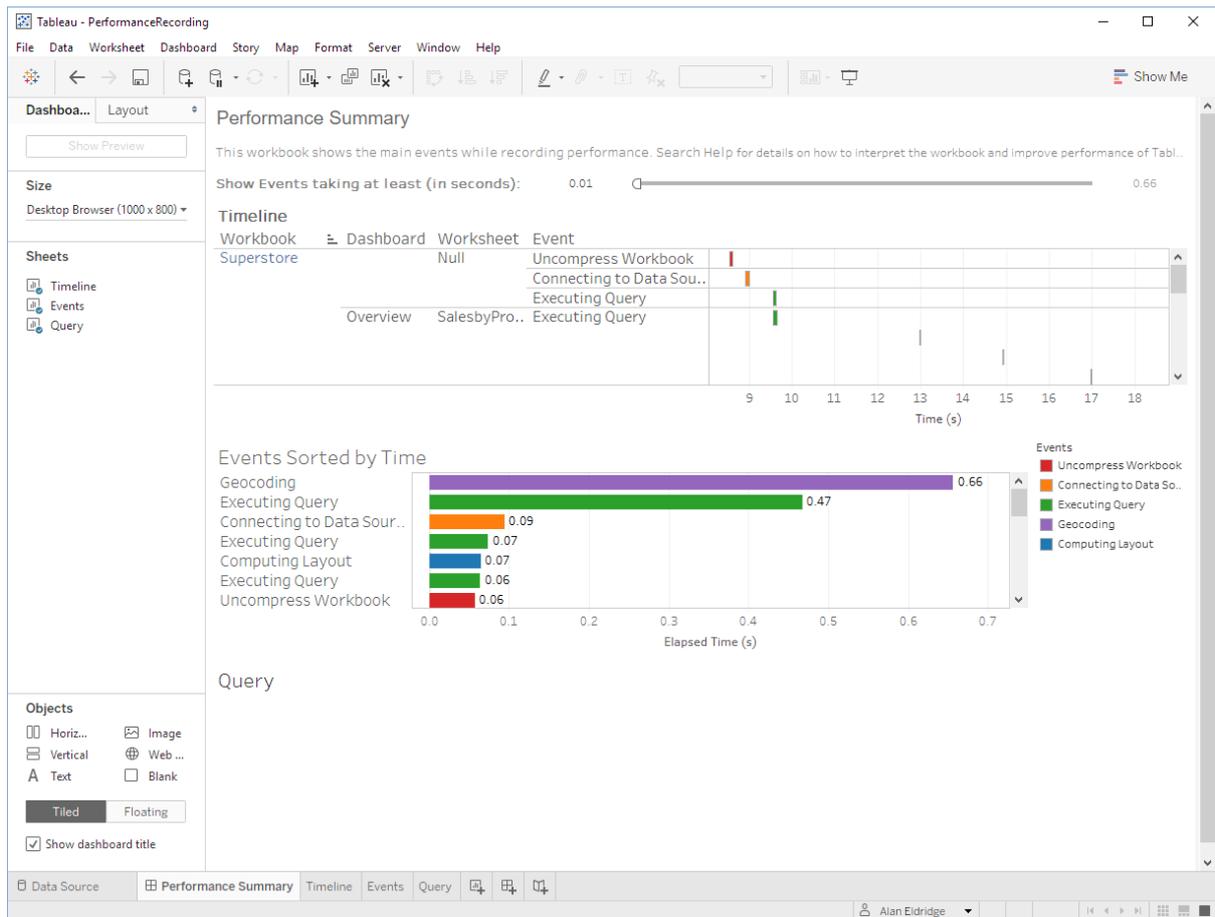
Um die Leistung Ihrer Arbeitsmappen zu verstehen, müssen Sie verstehen, a) was geschieht und b) wie lange es dauert. Diese Informationen werden an verschiedenen Stellen erfasst, je nachdem, wo Sie die Arbeitsmappe ausführen (d. h. in Tableau Desktop oder Tableau Server) und auf verschiedenen Detailebenen. In diesem Abschnitt werden die verschiedenen verfügbaren Optionen erörtert.

Leistungsaufzeichnung

Die erste Stelle, an der Sie Leistungsdaten ansehen sollten, ist die Funktion „Leistungsaufzeichnung“ von Tableau Desktop und Tableau Server. In Tableau Desktop aktivieren Sie diese Funktion im Hilfemenü:



Fahren Sie Tableau hoch und starten Sie die Leistungsaufzeichnung. Öffnen Sie Ihre Arbeitsmappe (es wird empfohlen, dabei keine weiteren Arbeitsmappen zu öffnen, damit kein Wettbewerb um Ressourcen entsteht). Interagieren Sie mit ihr so, als wären Sie ein Endbenutzer. Wenn Sie das Gefühl haben, dass Sie genügend Leistungsdaten gesammelt haben, wechseln Sie zurück zum Hilfemenü und stoppen Sie die Aufzeichnung. Ein weiteres Tableau Desktop-Fenster wird zu diesem Zeitpunkt mit den erfassten Daten geöffnet:



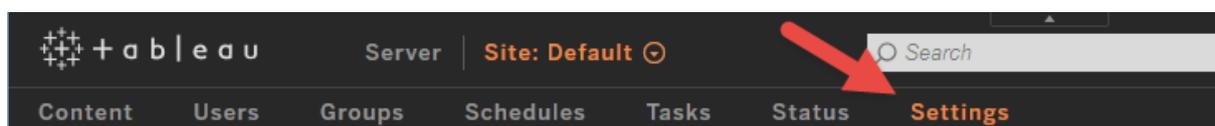
Sie können jetzt die Aktionen in der Arbeitsmappe erkennen, die die meiste Zeit in Anspruch nehmen. Beispielsweise benötigt die markierte Abfrage im Arbeitsblatt „Zeitachse“ im oben angezeigten Bild 30,66 Sekunden. Wenn Sie auf die Leiste klicken, wird der Text der Abfrage angezeigt, die ausgeführt wird. Da die Ausgabe der Leistungsaufzeichnung eine Tableau-Arbeitsmappe ist, können Sie auch zusätzliche Ansichten erstellen, um diese Daten auf andere Weise zu untersuchen.

Hinweis: Standardmäßig werden Ereignisse, die < 0,1 Sek. dauern, nicht angezeigt. Sie können diese einbeziehen, indem Sie den Filter oben im Dashboard anpassen, jedoch sollten Sie sich auf lang dauernde Aufgaben konzentrieren. Es wird empfohlen, diesen Wert auf 1 Sekunde zu setzen.

Es ist auch möglich, Leistungsaufzeichnungen auf Tableau Server durchzuführen, um Probleme zu untersuchen, die beim Veröffentlichen einer Arbeitsmappe entstehen. Die Leistungsaufzeichnung ist standardmäßig auf Tableau Server nicht aktiviert. Dies ist eine Funktion, die pro Site gesteuert werden kann.

Ein Server-Administrator kann die Leistungsaufzeichnung für einzelne Sites aktivieren.

1. Navigieren Sie zu der Site, für die die Leistungsaufzeichnung aktiviert werden soll.
2. Klicken Sie auf **Einstellungen**:



3. Wählen Sie unter „Arbeitsmappen-Leistungskennzahlen“ die Option **Arbeitsmappen-Leistungskennzahlen aufzeichnen**.

4. Klicken Sie auf **Speichern**.

Wenn Sie eine Leistungsaufzeichnung erstellen wollen:

1. Öffnen Sie die Ansicht, für die Sie eine Leistungsaufzeichnung starten wollen. Wenn Sie eine Ansicht öffnen, fügt Tableau Server der URL „:iid=<n>“ hinzu. Dies ist eine Session-ID. Beispiel:

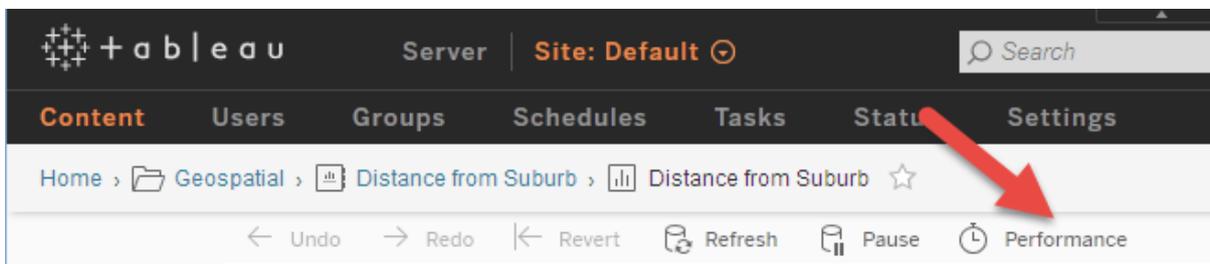
```
http://<tableau_server>/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?iid=1
```

2. Geben Sie am Ende der URL der Ansicht ein, unmittelbar vor der Session-ID. Beispiel:

```
http://<tableau_server>/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?record_performance=yes&iid=1
```

3. Laden Sie die Ansicht.

Eine visuelle Bestätigung, dass die Leistungsaufzeichnung gestartet wurde, ist die Option **Leistung** in der Symbolleiste der Ansicht:



Wenn Sie fertig sind und die Leistungsaufzeichnung ansehen wollen:

1. Klicken Sie auf **Leistung**, damit eine Leistungsarbeitsmappe geöffnet wird. Dabei handelt es sich um eine minutengenaue Momentaufnahme der Leistungsdaten. Sie können zusätzliche Momentaufnahmen anfertigen, während Sie mit der Ansicht arbeiten. Die Leistungsdaten sind kumulativ.
2. Wechseln Sie zu einer anderen Seite oder entfernen Sie von der URL, damit die Aufzeichnung gestoppt wird.

Sie sollten diese Information verwenden, um jene Bereiche einer Arbeitsmappe zu erkennen, bei denen sich eine Untersuchung lohnt, also wo Sie die meisten Verbesserungen für die aufgewendete Zeit erreichen können. Weitere Informationen zur Interpretation dieser Aufzeichnungen finden Sie unter dem folgenden Link:

http://onlinehelp.tableau.com/current/pro/desktop/de/de/help.htm#perf_record_create_desktop.html

Protokolle

In Tableau können Sie den vollständigen Text der Abfragen anzeigen, indem Sie sich die Protokolldatei ansehen. Der Standardspeicherort ist `C:\Users\\Documents\My Tableau Repository\Logs\log.txt`. Diese Datei ist sehr ausführlich und enthält JSON-codierten Text. Es wird also ein guter Texteditor wie etwa Notepad++ oder Sublime zur Bearbeitung empfohlen. Wenn Sie nach „begin-query“ oder „end-query“ suchen, können Sie die Abfragezeichenfolge finden, die an die Datenquelle weitergeleitet wird. Unter dem Protokolleintrag „end-query“ steht auch die Zeit, die für die Ausführung der Abfrage erforderlich war und wie viele Datensätze an Tableau zurückgegeben worden:

```

{"ts":"2015-05-24T12:25:41.226","pid":6460,"tid":"1674","sev":"info","req":"-
","sess":"-","site":"-","user":"-","k":"end-query",
"v":{"protocol":"4308fb0","cols":4,"query":"SELECT
[DimProductCategory].[ProductCategoryName] AS
[none:ProductCategoryName:nk],\n
[DimProductSubcategory].[ProductSubcategoryName] AS
[none:ProductSubcategoryName:nk],\n SUM(CAST(([FactSales].[ReturnQuantity])
as BIGINT)) AS [sum:ReturnQuantity:ok],\n SUM([FactSales].[SalesAmount]) AS
[sum:SalesAmount:ok]\nFROM [dbo].[FactSales] [FactSales]\n INNER JOIN
[dbo].[DimProduct] [DimProduct] ON ([FactSales].[ProductKey] =
[DimProduct].[ProductKey])\n INNER JOIN [dbo].[DimProductSubcategory]
[DimProductSubcategory] ON ([DimProduct].[ProductSubcategoryKey] =
[DimProductSubcategory].[ProductSubcategoryKey])\n INNER JOIN
[dbo].[DimProductCategory] [DimProductCategory] ON
([DimProductSubcategory].[ProductCategoryKey] =
[DimProductCategory].[ProductCategoryKey])\nGROUP BY
[DimProductCategory].[ProductCategoryName],\n
[DimProductSubcategory].[ProductSubcategoryName]","rows":32,"elapsed":0.951}}

```

Auf Tableau Server befinden sich die Protokolle in C:\ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Loggs.

Tableau Server Leistungsansichten

Tableau Server umfasst einige Ansichten für Administratoren, die die Überwachung der Tableau Server-Aktivitäten erleichtern sollen. Diese Ansichten befinden sich in der Tabelle „Analyse“ auf der Wartungsseite des Servers.

Analysis	
Dashboards that monitor Tableau Server activity.	
Dashboard	Analysis
Traffic to Sheets	Usage and users for published sheets.
Traffic to Data Sources	Usage and users for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	Sheet load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.
Server Disk Space	Current and historical disk space usage, by server node.
Tableau Desktop License Usage	Summary of usage for Tableau Desktop licenses
Tableau Desktop License Expirations	Expiration information for Tableau Desktop licenses

Weitere Informationen zu diesen Ansichten finden Sie unter dem folgenden Link:

<http://onlinehelp.tableau.com/current/server/de-de/adminview.htm>

Zusätzliche benutzerdefinierte Verwaltungsansichten können Sie erstellen, indem Sie eine Verbindung zur Datenbank PostgreSQL herstellen, einem Teil des Tableau-Repositorys. Anleitungen finden sich hier:

http://onlinehelp.tableau.com/current/server/de-de/adminview_postgres.htm

Überwachen und testen

TabMon

TabMon ist ein Open-Source-Cluster-Monitor für Tableau Server, mit dem Sie Leistungsstatistiken im Zeitverlauf sammeln können. Support für TabMon gibt es in der Community und unter der MIT-Open-Source-Lizenz geben wir den vollständigen Quellcode heraus.

TabMon erfasst in der Standardeinstellung den Systemzustand und Anwendungskennzahlen. Er sammelt integrierte Kennzahlen wie Windows Perfmon, Java Health und Java Mbean-Zähler (JMX) auf Tableau Server-Rechnern in einem Netzwerk. Sie können TabMon verwenden, um Komponenten- (CPU, RAM), Netzwerk- und Festplattennutzung zu überwachen. Sie können diese Werte verfolgen: Cache-Trefferquote, Anfragelatenz, aktive Sitzungen und vieles mehr. Die Daten werden in einer übersichtlichen, einheitlichen Struktur dargestellt, sodass es einfach ist, die Daten in Tableau Desktop zu visualisieren.

Mit TabMon haben Sie die vollständige Kontrolle darüber, welche Kennzahlen erfasst und welche Geräte überwacht werden sollen, ohne Skripte oder Code schreiben zu müssen. Sie brauchen lediglich die Bezeichnung des Geräts und der Kennzahl zu kennen. TabMon kann entfernt und unabhängig von Ihrem Cluster ausgeführt werden. Sie können den Zustand Ihrer Cluster von jedem beliebigen Computer in Ihrem Netzwerk aus fast ohne zusätzliche Datenverkehrslast auf Ihren Produktionsmaschinen überwachen, aggregieren und analysieren.

Weitere Informationen über TabMon finden Sie hier:

<http://bit.ly/1ULFelf>

TabJolt

TabJolt ist ein Tool zum Testen der Datenverkehrslast und der Leistung, speziell konzipiert für die einfache Arbeit mit Tableau Server. Es wird einfach auf einen Aspekt ausgerichtet und ausgeführt. Anders als herkömmliche Belastungstest-Tools kann TabJolt Datenverkehrslasten automatisch zu Ihrem Tableau Server leiten, ohne dass Skripte entwickelt oder gewartet werden müssen. Da TabJolt das Präsentationsmodell von Tableau kennt, kann es während der Testausführung automatisch Visualisierungen laden und mögliche Interaktionen interpretieren.

So können Sie TabJolt auf eine Ihrer Arbeitsmappen auf Ihrem Tableau Server ansetzen und Interaktionen mit den Tableau-Ansichten automatisch laden und ausführen. TabJolt sammelt auch wichtige Kennzahlen wie durchschnittliche Reaktionszeit, Durchsatz und 95.-Perzentil-Reaktionszeit und erfasst Windows-Leistungskennzahlen, um Korrelationen zu ermitteln.

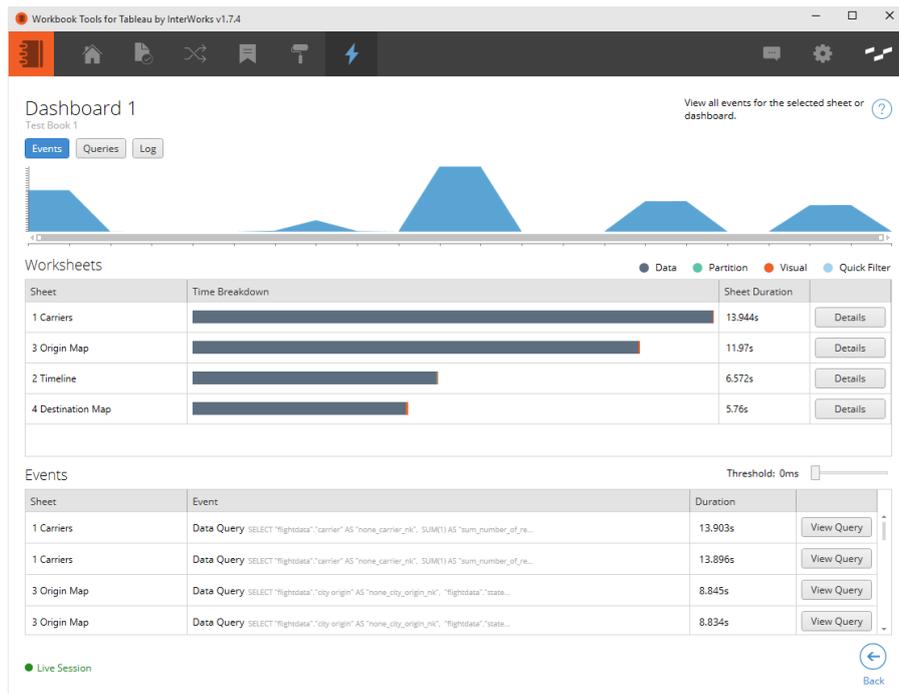
Natürlich müssen die Benutzer auch mit TabJolt über ausreichende Kenntnisse der Architektur von Tableau Server verfügen. Tableau Server wie eine Blackbox für Belastungstest zu behandeln, wird nicht empfohlen. Auf diese Weise würden Sie Ergebnisse erhalten, die nicht Ihren Erwartungen entsprechen. TabJolt ist ein automatisiertes Tool, mit dem nicht einfach die ganze Bandbreite menschlicher Interaktionen repliziert werden kann. Untersuchen Sie also sorgfältig, wie realistisch Ihre TabJolt-Ergebnisse sind.

Weitere Informationen über TabJolt finden Sie hier:

<http://bit.ly/1ULFtgi>

Weitere Tools

Es gibt einige andere Tools externer Anbieter, die Ihnen helfen, die Leistungsmerkmale Ihrer Arbeitsmappen zu ermitteln. Eine Option ist „Power Tools for Tableau“ von Interworks, das eine Leistungsanalyse umfasst (ähnlich wie die integrierte Leistungsaufzeichnung). Hiermit können Sie Detaildaten aufrufen, um zu verstehen, welche Blätter und Abfragen am meisten Zeit in Anspruch nehmen.



Palette Software bietet ebenfalls ein Produkt zur Erfassung von Leistungsdaten von Tableau Server. Es heißt Palette Insight. Mit diesem Tool können Sie Kapazitäten planen, Ressourcen-intensive Benutzer und Arbeitsmappen ermitteln, Benutzerzugriffe überwachen und Chargeback-Modelle aufbauen.



Außerdem beinhalten die meisten modernen DBMS-Plattformen Admin-Tools, mit deren Hilfe laufende Abfragen verfolgt und analysiert werden können. Ihr DBA kann Ihnen sicher helfen, falls Ihre Leistungsaufzeichnungen darauf hinweisen, dass die Laufzeit von Abfragen ein wesentlicher Faktor ist.

Wenn Sie der Meinung sind, dass die Interaktion zwischen dem Client-Browser und dem Tableau Server ein Problem ist, verschaffen Sie sich mit Tools wie Telerik Fiddler oder den Entwickler-Tools Ihres Browsers einen tiefer greifenden Einblick in den Datenverkehr zwischen Client und Tableau Server.

Liegt es am Design meiner Arbeitsmappe?

Das Arbeiten mit Tableau ist für viele Benutzer eine neue Erfahrung. Für die Erstellung effizienter Arbeitsmappen gibt es Designtechniken und bewährte Verfahrensweisen, die erlernt werden können. Wir haben jedoch beobachtet, dass viele Benutzer versuchen, alte Konzepte in Tableau anzuwenden, und dadurch viel Potenzial verschenken. In diesem Abschnitt werden einige empfohlene Designprinzipien behandelt.

Gutes Dashboard-Design

Mit Tableau erstellen Sie eine interaktive Erfahrung für Ihre Benutzer. Das Endergebnis von Tableau Server ist eine interaktive Anwendung, mit deren Hilfe die Benutzer die Daten erforschen können, anstatt sie nur zu betrachten. Um ein effizientes Tableau-Dashboard zu erstellen, müssen Sie also aufhören, in den Bahnen der Entwicklung eines statischen Berichts zu denken.

Es folgt ein Beispiel für einen Dashboard-Typ, der bei vielen neuen Autoren vorkommt, insbesondere, wenn sie zuvor mit Excel oder Access oder mit „traditionellen“ Berichtstools gearbeitet haben. Sehen Sie sich den folgenden Tabellenbericht an, der „alles“ plus eine Serie von Filtern enthält, mit deren Hilfe der Benutzer die Tabelle verfeinern kann, bis sie diejenigen Datensätze enthält, die ihn interessieren:

A Bad Design									
Continent	Country	State/Province	Product Category	Product Subcate.	Product Name	Sales Qty	Total Cost	Sales Amou..	
Asia	Turkmenistan	Ahal Province	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	64	\$5,547.52	\$11,790.68	
North America	United States	Alaska	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	62	\$5,200.80	\$11,536.20	
North America	Canada	Alberta	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	40	\$3,467.20	\$7,540.00	
Europe	France	Alpes-Maritim.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	31	\$2,687.08	\$5,734.17	
Asia	Armenia	Armenia	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	19	\$1,560.24	\$3,468.40	
Europe	France	Bas-Rhin	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	36	\$3,033.80	\$6,710.60	
Europe	Germany	Bavaria	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	60	\$5,114.12	\$10,819.90	
Asia	China	Beijing	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	2,276	\$194,683.28	\$421,825.30	
Europe	Germany	Berlin	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	1,193	\$102,022.36	\$220,330.11	
Europe	Switzerland	Bern	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	38	\$3,207.16	\$6,936.80	
North America	Canada	British Colum.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	107	\$9,274.76	\$20,075.25	
Europe	Romania	Bucuresti	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	10	\$866.80	\$1,885.00	
Europe	Greece	Central Greec.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	18	\$1,560.24	\$3,317.60	
Asia	Japan	Chubu	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	13	\$1,040.16	\$2,337.40	
Asia	Kyrgyzstan	Chuy Province	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	66	\$5,547.52	\$12,280.78	
North America	United States	Colorado	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	699	\$59,635.84	\$130,093.28	
North America	United States	Connecticut	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	293	\$25,050.52	\$54,533.05	
Asia	Syria	Damascus	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	104	\$8,928.04	\$19,311.83	
Europe	United Kingdo.	England	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	478	\$40,912.96	\$88,702.45	
North America	United States	Florida	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	423	\$35,885.52	\$78,745.88	
Europe	Germany	Hesse	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	26	\$2,253.68	\$4,674.80	
Asia	Japan	Hokkaido	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	22	\$1,906.96	\$4,109.30	
Asia	China	Hong Kong	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	111	\$9,448.12	\$20,574.78	
Asia	Pakistan	Islamabad Ca.	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	63	\$5,287.48	\$11,724.70	
Asia	Japan	Kansai	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	119	\$10,228.24	\$22,158.18	
Asia	Japan	Kanto	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	176	\$15,255.68	\$32,648.20	
Asia	Thailand	Krung Thep	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	89	\$7,541.16	\$16,522.03	
Europe	Ireland	Leinster	Cameras and cam.	Digital Cameras	A. Datum Advanced Digital Ca..	31	\$2,687.08	\$5,824.65	

Continent

 Asia
 Europe
 North America

Country

 Armenia
 Australia
 Bhutan
 Canada

City

 Null
 Albany
 Alexandria
 Amsterdam

Product Category

 Audio
 Cameras and camcorders
 Cell phones
 Computers

Product Subcategory

 Air Conditioners
 Bluetooth Headphones
 Boxed Games
 Camcorders

Product Name

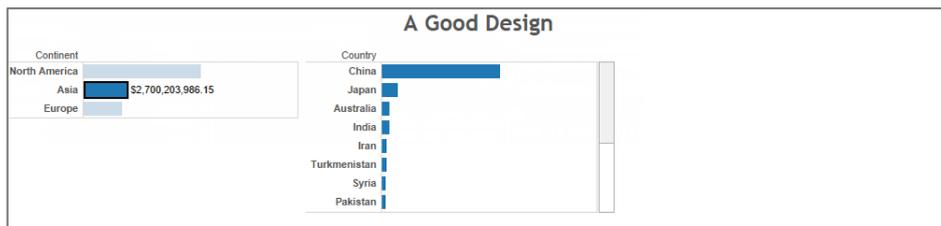
 A. Datum Advanced Digi.
 A. Datum Advanced Digi.
 A. Datum Advanced Digi.
 A. Datum Advanced Digi.

Dies ist kein „gutes“ Tableau-Dashboard (eigentlich ist es überhaupt kein „gutes“ Dashboard). Vielleicht ist es eher ein verbessertes Verfahren zum Extrahieren von Daten, weil der Benutzer die Daten in ein anderes Tool wie Excel übertragen möchte, um sie weiter zu analysieren und Diagramme zu erstellen. Vielleicht weist es auch gerade darauf hin, dass gar nicht bekannt ist, wie der Benutzer letztendlich die Daten untersuchen möchte. Der Ansatz ist also: „Ausgehend von den Anfangskriterien sehen Sie hier alles ... Und hier haben Sie einige Filterobjekte, damit Sie den Ergebnisdatensatz verfeinern und letztendlich finden können, was sie eigentlich suchen.“

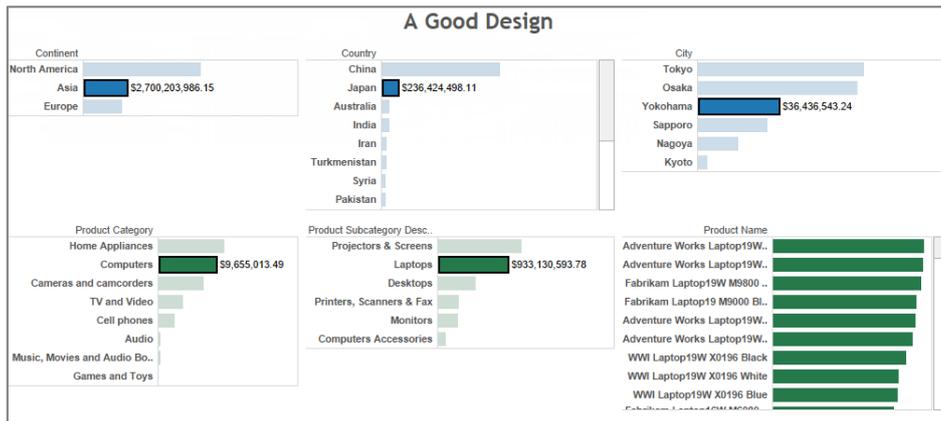
Jetzt schauen Sie sich die folgende Aufarbeitung an, bei der es sich um genau die gleichen Daten handelt. Hier beginnen Sie mit der höchsten Aggregationsebene:



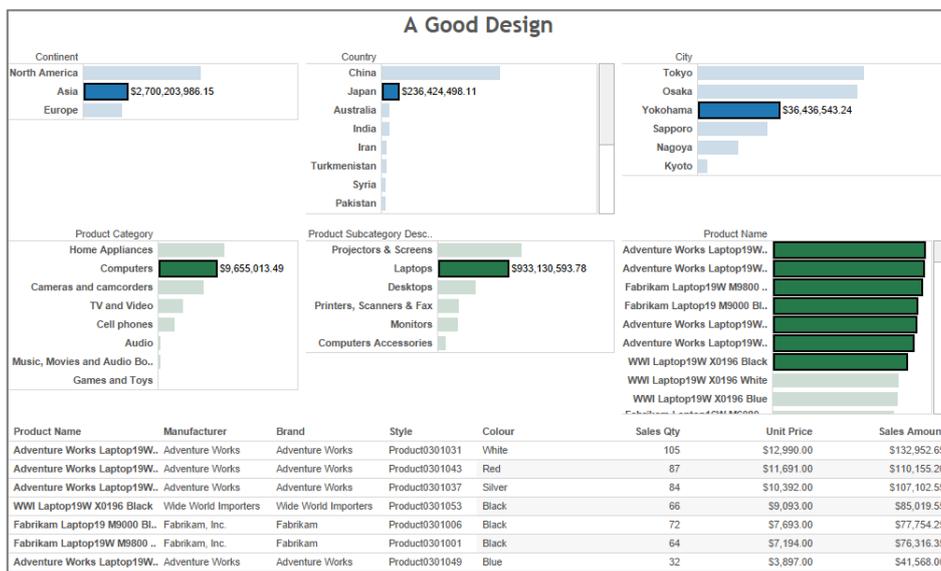
Wenn Sie eines oder mehrere Elemente auswählen, wird die nächste Detailebene angezeigt:



Wenn Sie fortfahren, werden jedes Mal mehr Details angezeigt:



Bis Sie schließlich auf der letzten Ebene angelangt sind. Die gleichen Daten, die oben im Kreuztabellen-Dashboard gezeigt wurden.



Konzentrieren Sie sich nicht auf die Präsentation der Daten (das ist wichtig, aber darauf gehen wir später näher ein). Denken Sie stattdessen an die Erfahrung bei der Benutzung dieses Dashboards. Beachten Sie, wie es an einem natürlichen Pfad entlang fließt, von links nach rechts, von oben nach unten. Diesem Beispiel können sehr viele Daten zugrunde liegen, aber das Dashboard führt den Endbenutzer stufenweise zu weiteren Detailebenen, bis er schließlich die gesuchten Detaildaten findet.

Der wesentliche Unterschied zwischen diesen beiden Beispielen liegt darin, wie sie den Endbenutzer durch den analytischen Prozess führen. Das erste Beispiel beginnt breit (es werden so viele

Datensätze gezeigt, wie man anschauen kann), dann reduziert der Endbenutzer die Anzahl der angezeigten Datensätze, indem er Filter anwendet. Diese Technik birgt jedoch Probleme in sich:

- Die ursprüngliche Abfrage, die ausgeführt werden muss, bevor dem Endbenutzer alles gezeigt wird, ist möglicherweise die größte Abfrage, die sie stellen können: „Gib mir alle Datensätze“. Bei einem realistischen Datensatz nimmt die Ausführung und Rücksendung zur Tableau-Engine wahrscheinlich geraume Zeit in Anspruch. Der „erste Eindruck“ ist wichtig für die Wahrnehmung der Lösung durch den Endbenutzer. Wenn es länger als ein paar Sekunden dauert, bevor etwas geschieht, ist die Wahrnehmung gleich negativ.
- Eine Ansicht mit hunderttausenden oder Millionen von Zellen zu erstellen (jede Zelle in einer Kreuztabelle wird als „Markierung“ bezeichnet), erfordert sehr viel Prozessorleistung und Arbeitsspeicher. Es erfordert auch Zeit, was die negative Wahrnehmung der Systemreaktion noch verstärkt. Wenn zahlreiche Personen auf Tableau Server große Kreuztabellen generieren, kann die Leistung sinken und das System hat im schlimmsten Fall keinen Speicher mehr. Dies kann Probleme mit der Serverstabilität verursachen, auch Fehler und verschiedenste Arten von unangenehmen Benutzererfahrungen. Natürlich können Sie auf dem Server mehr Speicher hinzufügen, um diese Auswirkungen zu minimieren, aber damit wird das Symptom behandelt, nicht die Ursache.
- Schließlich wird den Benutzern nicht kontextbezogen klargemacht, ob die anfänglich verfügbaren Filter zu weit oder zu eng gefasst sind. Wie soll der Benutzer eines Berichts wissen, dass zehntausende Datensätze zurückgegeben werden, wenn alle verfügbaren Kategorien aktiviert werden und so der gesamte verfügbare Arbeitsspeicher auf dem Server erschöpft wird? Das wissen sie erst nach der schmerzlichen Erfahrung.

Betrachten Sie im Gegensatz dazu den zweiten Ansatz, wo die erste Abfrage nur die höchste Aggregationsebene anzeigt:

- Die anfänglich auszuführende Abfrage ist hochgradig aggregiert und gibt somit nur einige wenige Datensätze zurück. Bei einer gut aufgebauten Datenbank ist dies ein sehr effizienter Vorgang, somit ist die Reaktionszeit beim „ersten Eindruck“ sehr kurz und hat eine positive Systemwahrnehmung zur Folge. Die folgenden detaillierteren Ebenen sind sowohl aggregiert als auch eingeschränkt durch die Auswahl auf der höheren Ebene. Sie lassen sich weiterhin schnell ausführen und liefern die Daten schnell an die Tableau-Engine zurück.
- Wenn das Dashboard vollständig ist, haben Sie zwar mehr Ansichten, jedoch enthält jede Ansicht nur einige Dutzend Zellen. Die zum Generieren dieser Ansichten erforderlichen Ressourcen sind vernachlässigbar, auch wenn viele Endbenutzer im System aktiv sind, und die Wahrscheinlichkeit, dass der Systemspeicher unzureichend ist, bleibt gering.
- Schließlich können Sie sehen, dass für die höheren „Navigationsebenen“ die Umsätze der einzelnen Kategorien gezeigt werden. Dies vermittelt dem Benutzer einen Kontext, aus dem ersichtlich ist, ob diese Auswahl viele oder wenige Datensätze enthält. Außerdem wurden Farben verwendet, um auf die Wirtschaftlichkeit der einzelnen Kategorien hinzuweisen. Dies gewinnt insofern an Bedeutung, als der Benutzer nun sieht, welche spezifischen Bereiche Aufmerksamkeit erfordern, anstatt nur blind zu navigieren.

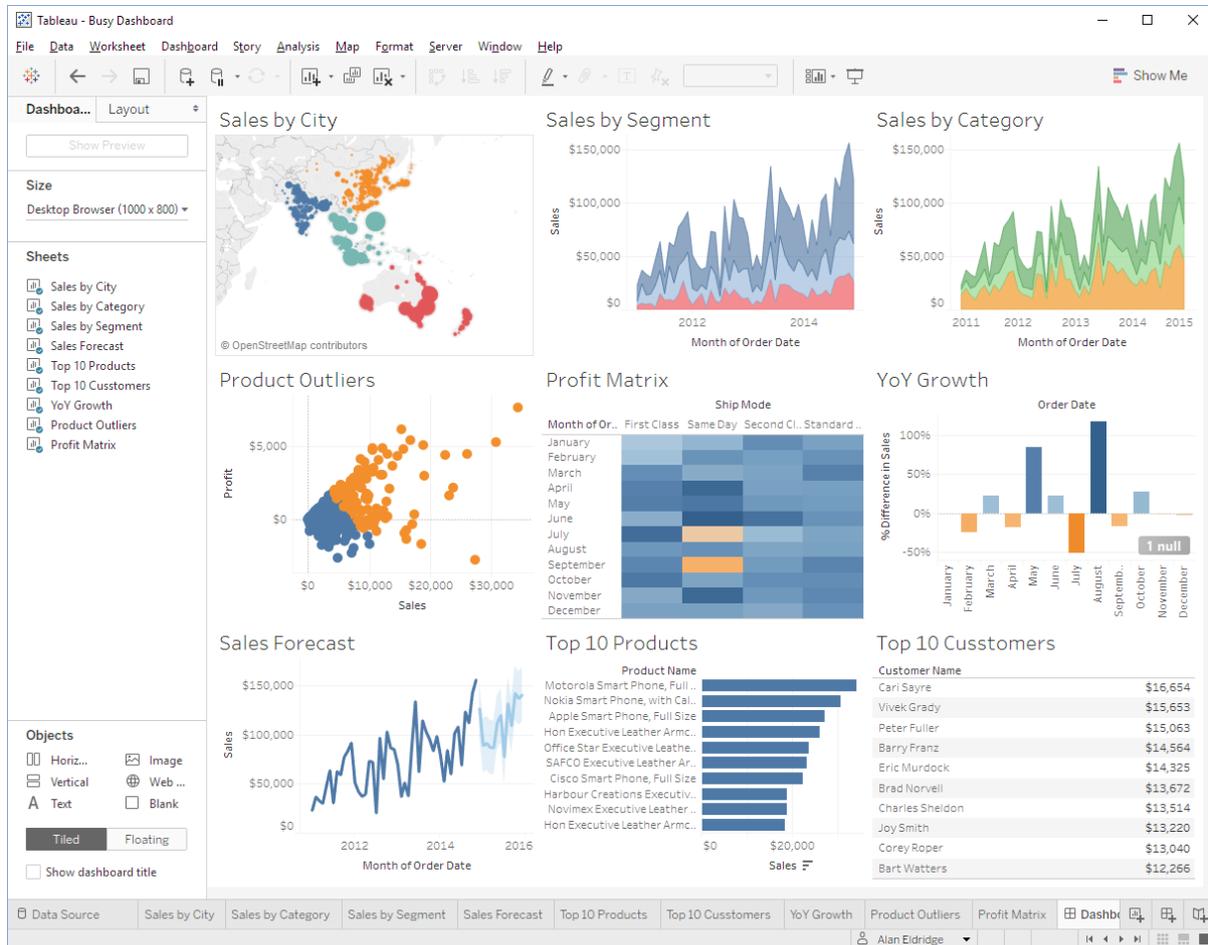
Keep it simple

Ein häufiger Fehler neuer Benutzer ist, dass sie Dashboards erstellen, die übermäßig komplex sind. Vielleicht versuchen sie ein Dokument neu zu erstellen, das sie bisher bei einem anderen Tool verwendet haben, oder sie versuchen etwas zu erstellen, das speziell für einen gedruckten Bericht konzipiert wurde. Das Endergebnis ist eine langsame und ineffiziente Arbeitsmappe.

Die folgenden Punkte tragen zur Komplexität bei:

Zu viele Arbeitsmappen pro Dashboard

Ein häufiger Fehler neuer Benutzer besteht darin, übermäßig viele Grafiken und Arbeitsmappen in einem Dashboard unterzubringen.



Bedenken Sie, dass jedes Arbeitsblatt eine (möglicherweise mehrere) Abfragen in den Datenquellen ausführen wird. Je mehr Arbeitsblätter, desto länger dauert es, das Dashboard zu rendern. Nutzen Sie die Tatsache, dass Tableau so konzipiert ist, dass es dem Endbenutzer interaktive Dashboards anbietet. Verteilen Sie die Daten also über mehrere Dashboards bzw. Seiten.

Zu viele Filterkarten

Filterkarten sind eine leistungsstarke Funktion von Tableau, mit deren Hilfe Sie reichhaltige, interaktive Dashboards für Endbenutzer erstellen können. Jedoch kann jeder Filter eine Abfrage erfordern, um die Optionen aufzuzählen. Wenn Ihrem Dashboard also zu viele Filter hinzugefügt werden, kann dies zur Folge haben, dass das Dashboard viel Zeit zum Rendern benötigt. Wenn Sie bei einem Filter „Relevante Werte anzeigen“ verwenden, müssen die angezeigten Werte mithilfe einer Abfrage aktualisiert werden, sobald andere Filter geändert werden. Setzen Sie diese Funktion sparsam ein.

Wenn Sie Ihre Filter auf mehrere Arbeitsblätter anwenden, achten Sie darauf, dass jede Änderung mehrere Abfragen auslöst, weil alle betroffenen, sichtbaren Arbeitsblätter aktualisiert werden (nicht sichtbare Arbeitsblätter werden nicht ausgeführt). Wenn dies einige Sekunden in Anspruch nimmt, hat es eine schlechte Nutzererfahrung zur Folge. Wenn Sie erwarten, dass die Benutzer an einem Filter mit Mehrfachauswahl mehrere Änderungen vornehmen, fügen Sie die Schaltfläche

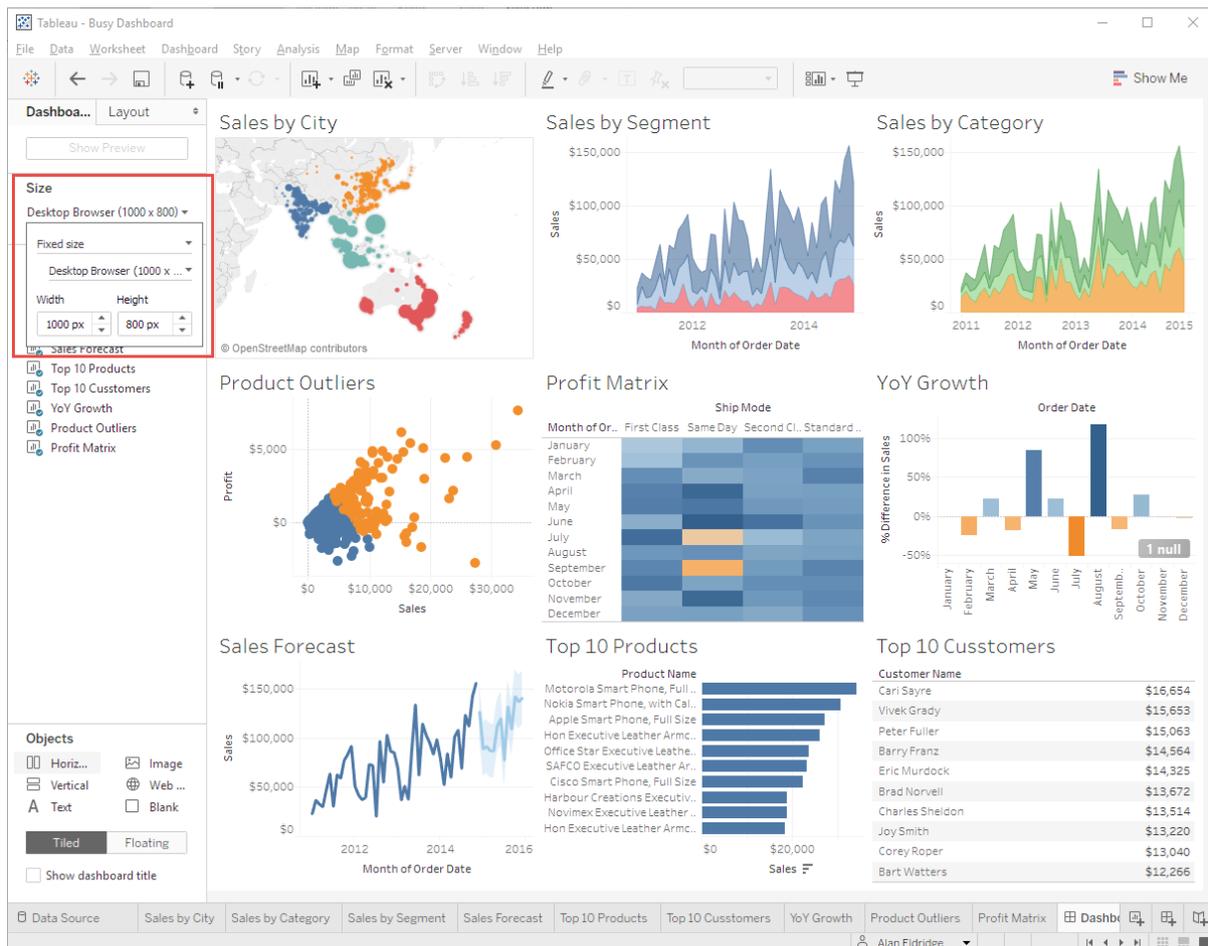
„Übernehmen“ hinzu. So kann der Benutzer die Aktualisierung auslösen, wenn alle gewünschten Optionen ausgewählt sind.

Tunen Sie Ihr Dashboard auf Leistung

Wenn Sie sichergestellt haben, dass Ihr Dashboard so einfach wie möglich ist, können Sie das Design feinabstimmen, damit der Cachespeicher für weitere Leistungsverbesserung genutzt wird.

Dashboards mit fester Größe

Eines der einfachsten Dinge zur Verbesserung der Leistung besteht darin, sicherzustellen, dass das Dashboard eine feste Größe hat.



Ein Teil des Rendering-Prozesses in Tableau erstellt ein Layout: wie viele Zeilen/Spalten für Rasterdiagramme und Kreuztabellen angezeigt werden sollen, Anzahl und Intervall der einzuziehenden Achsenteilstriche/Rasterlinien, Anzahl und Ort der angezeigten Markierungsbeschriftungen usw. Dies wird durch die Größe des Fensters bestimmt, in dem das Dashboard angezeigt werden soll.

Wenn das gleiche Dashboard mehrmals von Fenstern unterschiedlicher Größe angefordert wird, muss für jede Anfrage ein Layout generiert werden. Wenn für das Dashboard-Layout eine feste Größe definiert wird, stellen wir sicher, dass nur ein Layout erstellt werden muss, dass bei allen Anfragen wiederverwendet werden kann. Beim serverseitigen Rendern ist dies sogar noch wichtiger, weil bei Dashboards mit fester Größe auch die auf dem Server gerenderten Bitmaps im Cache abgelegt und freigegeben werden können. Dies verbessert sowohl die Leistung als auch die Skalierbarkeit.

Gerätespezifische Dashboards

In Tableau 10 führen wir eine neue Funktion mit der Bezeichnung gerätespezifische Dashboards ein. Mit dieser Funktion können Sie spezifische Dashboard-Layouts erstellen, die je nach dem verwendeten Gerät automatisch ausgewählt werden.

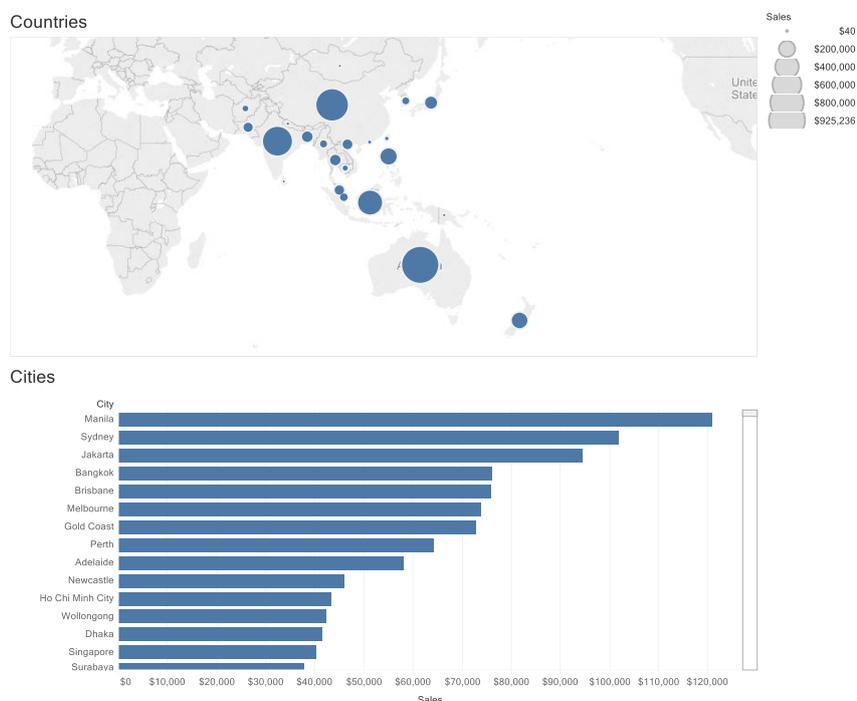
Wir wählen das zu verwendenden Layout gemäß der Bildschirmgröße:

- ≤ 500 px in der kürzesten Achse – Telefon
- ≤ 800 px in der kürzesten Achse – Tablet
- > 800 px – Desktop

Da die verschiedenen Geräte innerhalb dieser Klassen unterschiedliche Bildschirmgrößen haben und die Geräte gedreht werden können, sollten die Layouts für Telefone bzw. Tablets auf automatische Größenanpassung gesetzt werden. So hat der Benutzer die beste Erfahrung beim Betrachten auf verschiedenen Geräten. Dies wirkt sich jedoch auf die Wiederverwendung des Cache-Speichers aus (sowohl der Zwischenspeicher für das Präsentationsmodell als auch für Bildkacheln für das serverseitige Rendering). Im Allgemeinen ist der Vorteil der ordnungsgemäßen Größenanpassung an das jeweilige Gerät wichtiger als die Nutzung des Zwischenspeichers, man sollte sich dessen jedoch bewusst sein. Nachdem der Benutzer begonnen hat, eine Arbeitsmappe zu benutzen, werden die Modelle und Bitmaps für die gängigsten Bildschirmgrößen übernommen und damit die Leistung verbessert.

Nutzung der Detailgenauigkeit der Visualisierung zum Reduzieren von Abfragen

Obwohl die bewährten Verfahrensweisen im Allgemeinen darin bestehen, für jede Arbeitsmappe nur die benötigten Felder zu verwenden, können wir gelegentlich die Leistung verbessern, indem wir mehr Daten in ein Arbeitsblatt übernehmen, um gegenseitige Abfragen zu vermeiden. Betrachten Sie das folgende Dashboard:



Bei erwartungsgemäßem Aufbau würde der Ausführungsplanung zwei Abfragen ergeben – für jedes Arbeitsblatt eine:

Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10  2.11

Timeline

Workbook	Dashboard	Worksheet	Event	Time (s)
Book1	Dashboard 1	Countries	Executing Query	807 - 808
		Cities	Executing Query	813 - 815

```
SELECT [Superstore APAC].[Stadt] AS [Stadt],
       SUM([Superstore APAC].[Umsatz]) AS [sum:Umsatz:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Stadt]
```

```
SELECT [Superstore APAC].[Land] AS [Land],
       SUM([Superstore APAC].[Umsatz]) AS [sum:Umsatz:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Land]
```

Wenn wir unser Dashboard-Design ändern und „Land“ in das Arbeitsblatt „Städte“ übernehmen (im Detailcontainer), kann Tableau das Dashboard mit nur einer Abfrage vervollständigen. Tableau ist so intelligent, die Abfrage für das Arbeitsblatt „Städte“ zuerst auszuführen und dann den Cache-Speicher mit den Abfrageergebnissen zu nutzen, um die Daten für das Arbeitsblatt „Länder“ zu liefern. Diese Funktion wird „query batching“ (Abfrage Stapelung) genannt.

Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10  3.02

Timeline

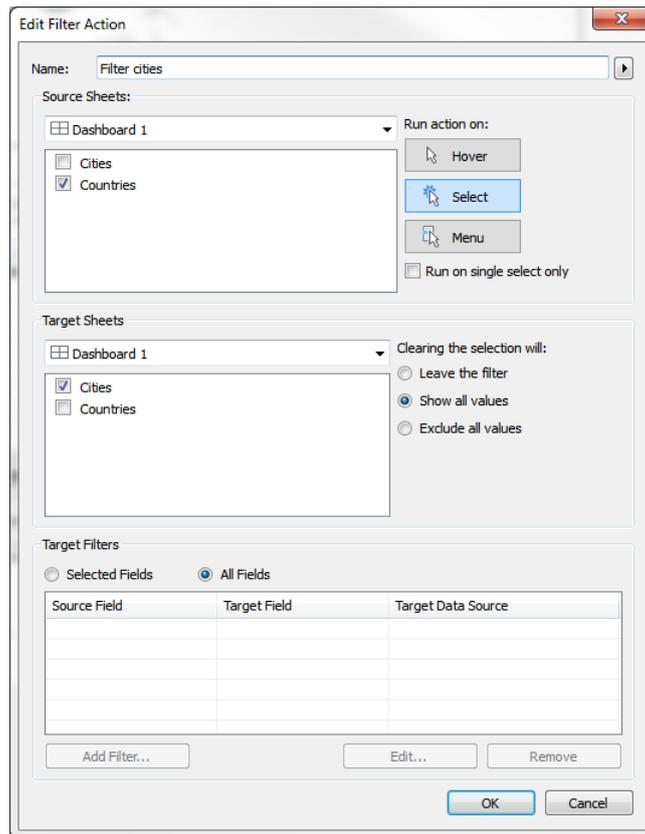
Workbook	Dashboard	Worksheet	Event	Time (s)
Book1	Dashboard 1	Cities	Executing Query	126.5 - 129.5

```
SELECT [Superstore APAC].[Stadt] AS [Stadt],
       [Superstore APAC].[Land] AS [Land],
       SUM([Superstore APAC].[Umsatz]) AS [sum:Umsatz:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Stadt],
         [Superstore APAC].[Land]
```

Dies ist natürlich nicht in allen Fällen möglich, weil das Hinzufügen einer Dimension zu einer Visualisierung die Detailgenauigkeit verändert, was zur Folge hat, dass mehr Markierungen dargestellt werden. Wenn Sie jedoch eine hierarchische Beziehung wie im obigen Beispiel in Ihren Daten haben, ist dies eine nützliche Technik, weil sie keine Auswirkung auf die sichtbare Detailgenauigkeit hat.

Nutzung der Detailgenauigkeit zum Optimieren von Aktionen

Bei Aktionen können wir auf ähnliche Weise vorgehen, um die Anzahl der auszuführenden Abfragen zu reduzieren. Sehen Sie sich dasselbe Dashboard oben an (vor der Optimierung). Jetzt fügen wir dem Arbeitsblatt „Städte“ eine Filteraktion vom Arbeitsblatt „Länder“ hinzu.



Wenn wir diese Aktion auslösen, indem wir auf eine Markierung auf der Karte klicken, sehen wir, dass Tableau eine Abfrage ausführen muss, um die Werte im Arbeitsblatt „Städte“ zu ermitteln. Der Grund dafür ist, dass der Cache-Speicher für das Abfrageergebnis keine Daten hinsichtlich der Stadt-Land-Beziehung enthält.

Performance Summary

This workbook shows the main events while recording performance. Search Help for details on how to interpret the workbook and improve performance of Tableau.

Show Events taking at least (in seconds): 0.10 13.47

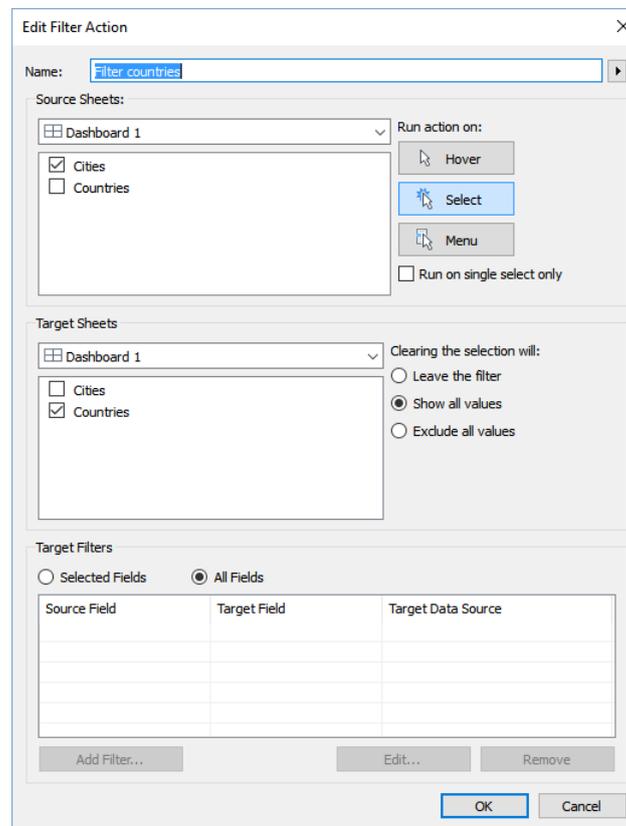
Timeline



```
SELECT [Superstore APAC].[Stadt] AS [Stadt],
       SUM([Superstore APAC].[Umsatz]) AS [sum:Umsatz:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Land] = 'Australien')
GROUP BY [Superstore APAC].[Stadt]
```

Wenn wir „Land“ zum Arbeitsblatt „Städte“ hinzufügen, haben wir jetzt ausreichende Angaben im Abfrageergebnis-Cache, um diese Filter anzuwenden, ohne erneut auf die Datenquelle zugreifen zu müssen.

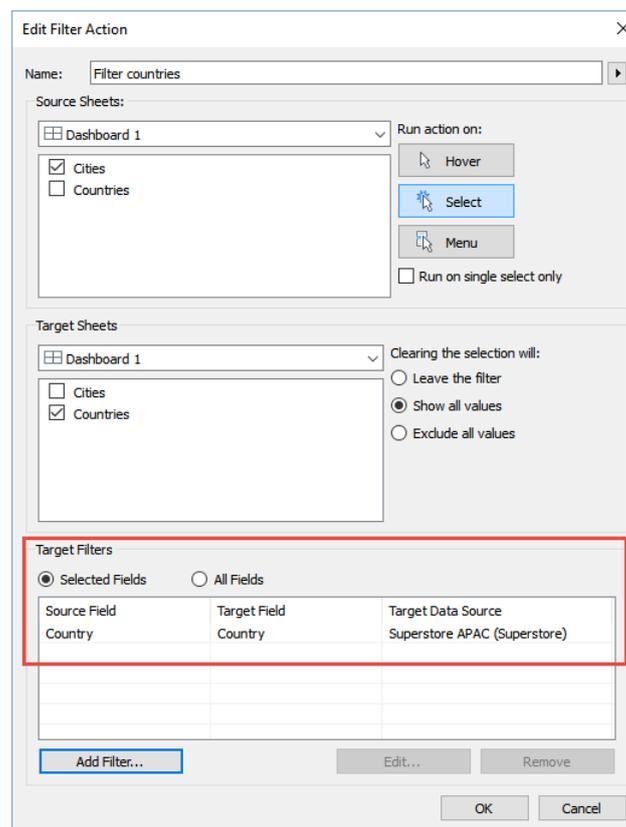
Auf ähnliche Weise können wir Situationen optimieren, wo das Quellarbeitsblatt detaillierter ist als das Ziel. Wenn wir die standardmäßige Aktionsdefinition verwenden würden, wo wir mit „allen Feldern“ filtern:



Das hat zur Folge, dass die Arbeitsmappe für jede Aktion eine Abfrage ausführt, weil die Filterklausel sich auf „Land“ und „Stadt“ bezieht, was nicht aus dem Abfrageergebnis-Cache für das Arbeitsblatt „Länder“ bedient werden kann.

```
SELECT [Superstore APAC].[Land] AS [Land],
       SUM([Superstore APAC].[Umsatz]) AS [sum:Umsatz:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE (([Superstore APAC].[Stadt] = 'Sydney') AND ([Superstore APAC].[Land] =
'Australien'))
GROUP BY [Superstore APAC].[Land]
```

Wenn wir die Aktion so ändern, dass nur auf der Grundlage von „Land“ gefiltert wird:



Jetzt können wir diesen Filter aus dem Abfrageergebnis-Cache bedienen, brauchen also keine Abfrage in der Datenquelle. Wie oben, müssen Sie überlegen, ob eine Veränderung der Detailgenauigkeit das Design Ihres Arbeitsplatzes beeinflusst. Wenn dies nicht der Fall ist, ist dieses Verfahren nützlich.

Gutes Arbeitsblattdesign

Die nächste Ebene unterhalb des Dashboards ist das Arbeitsblatt. In Tableau besteht eine enge Verbindung zwischen dem Arbeitsblattdesign und den auszuführenden Abfragen. Jedes Arbeitsblatt generiert eine oder mehrere Abfragen. Auf dieser Ebene müssen wir also gewährleisten, dass die generierten Abfragen möglichst optimal sind.

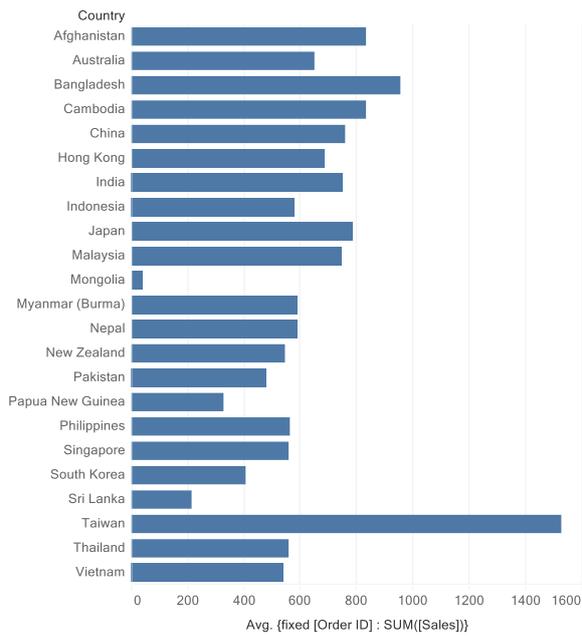
Schließen Sie nur benötigte Felder ein.

Sehen Sie sich den Detailcontainer an und entfernen Sie alle Felder, die nicht direkt in der Visualisierung verwendet, im QuickInfo benötigt werden oder für die Ausführung des geforderten Markierungsdetails erforderlich sind. Damit läuft die Abfrage in der Datenquelle schneller ab und es müssen weniger Daten als Abfrageergebnisse zurückgegeben werden. Es gibt einige Ausnahmen für diese Regel, die wir bereits untersucht haben (Abfrageverbindung zur Vermeidung ähnlicher Abfragen in anderen Arbeitsblättern). Sie treten jedoch weniger häufig auf und sind abhängig davon, dass die visuelle Detailgenauigkeit des Blatts nicht geändert wird.

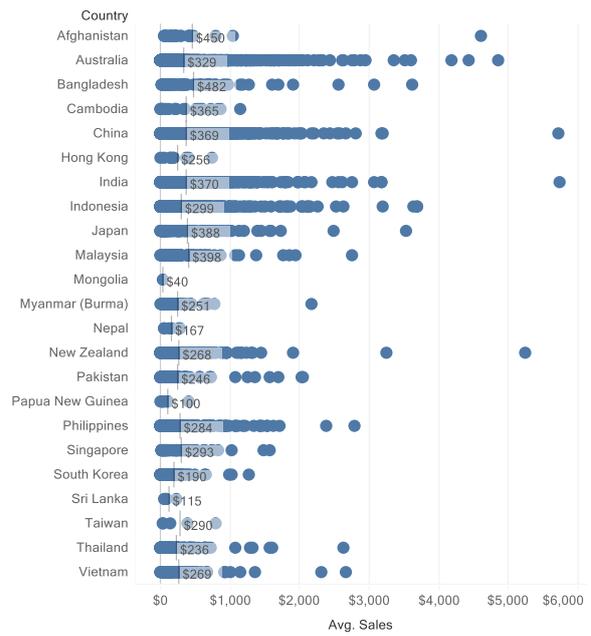
Zur Beantwortung der Frage so wenig Markierungen wie möglich anzeigen

In Tableau gibt es häufig mehrere Möglichkeiten, eine bestimmte Zahl zu berechnen. Betrachten Sie das folgende Dashboard. Beide Arbeitsblätter beantworten die Frage „Welches ist die mittlere Auftragsgröße pro Land?“

Avg Order Size 1



Avg Order Size 2

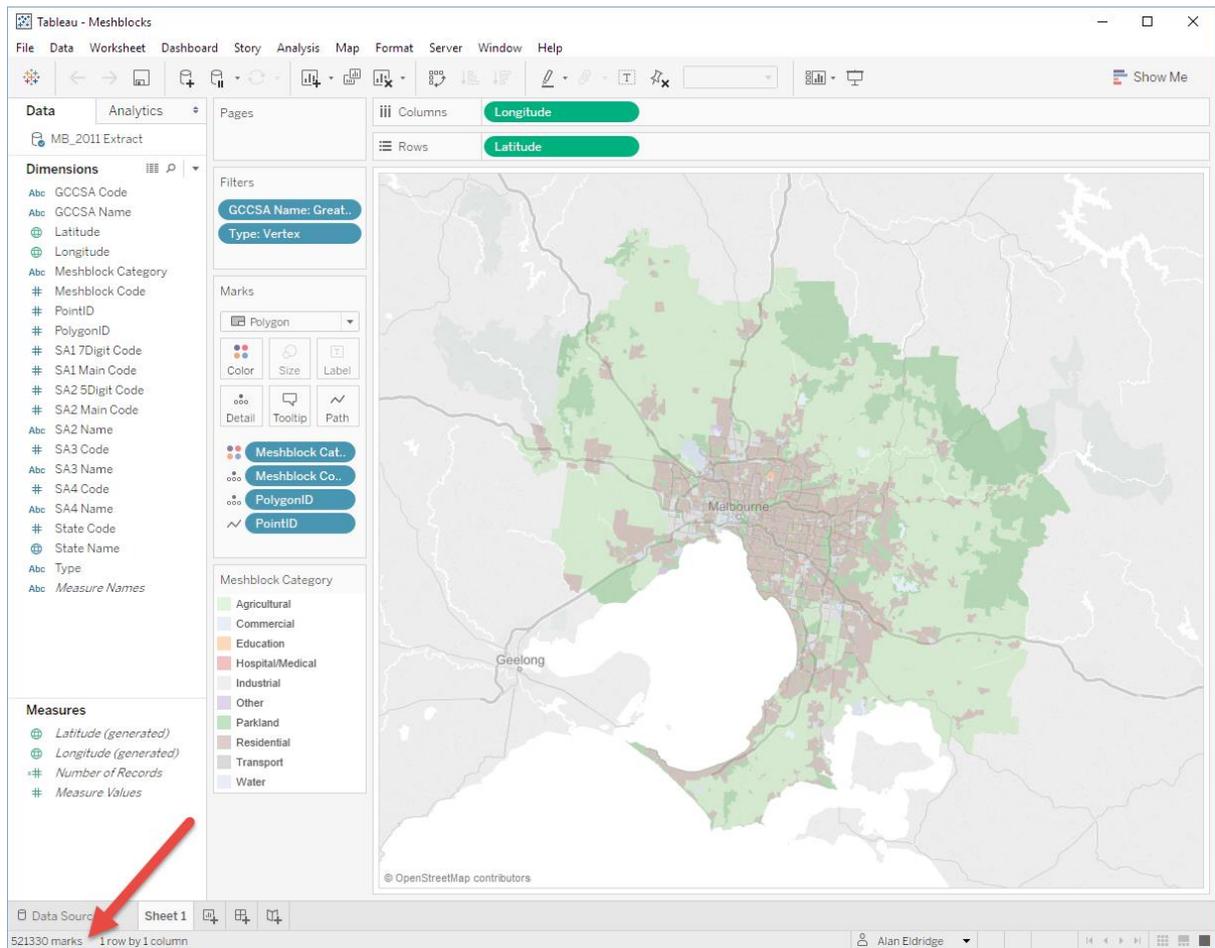


Blatt 1 enthält nur eine Markierung, die die mittlere Auftragsgröße für jedes Land repräsentiert. Es werden also nur 23 Datensätze von der Datenquelle zurückgegeben. In Blatt 2 jedoch wird für jeden Auftrag in jedem Land eine Markierung gezeigt, der Durchschnitt berechnet und als Referenzlinie angezeigt. Dafür müssen 5436 Datensätze aus der Datenquelle abgerufen werden.

Blatt 1 ist die bessere Lösung, wenn wir nur an der ursprünglichen Frage interessiert sind: Welches ist die mittlere Auftragsgröße pro Land? Blatt 2 dagegen beantwortet dies, bietet jedoch darüber hinaus auch einen tieferen Einblick in die Verteilung der Auftragsgrößen, sodass wir Ausreißer erkennen können.

Hochkomplexe Visualisierungen vermeiden

Eine wichtige Kennzahl, die wir beachten sollten, ist die Anzahl der Datenpunkte, die in jeder einzelnen Visualisierung gerendert werden. Dies können Sie leicht herausfinden, indem Sie die Statusleiste des Tableau Desktop-Fensters ansehen:

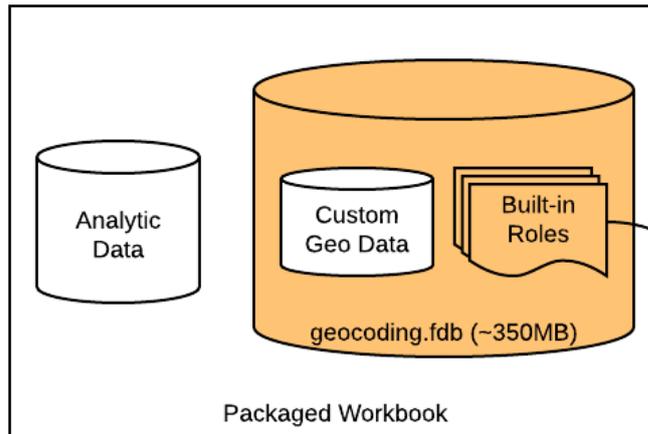


Es gibt zwar keine eindeutige Regel, was „zu viele Markierungen“ bedeutet, aber berücksichtigen Sie, dass mehr Markierungen zum Rendern mehr CPU und RAM benötigen. Achten Sie auf große Kreuztabellen, Streudiagramme und/oder Karten mit komplexen benutzerdefinierten Polygonen.

Karten

Benutzerdefinierte Geokodierung

Wenn Sie eine benutzerdefinierte Geokodierungsrolle importieren, wird diese in die Geokodierungsdatenbank geschrieben, eine Firebird DB-Datei, die standardmäßig in C:\Programme\Tableau\Tableau 10.0\Local\data\geocoding.fdb gespeichert ist. Wenn Sie die Rolle in einer Arbeitsmappe verwenden und als Arbeitsmappenpaket speichern, wird die gesamte Datenbankdatei in eine TWBX-Datei komprimiert – die gesamten rund 350 MB!

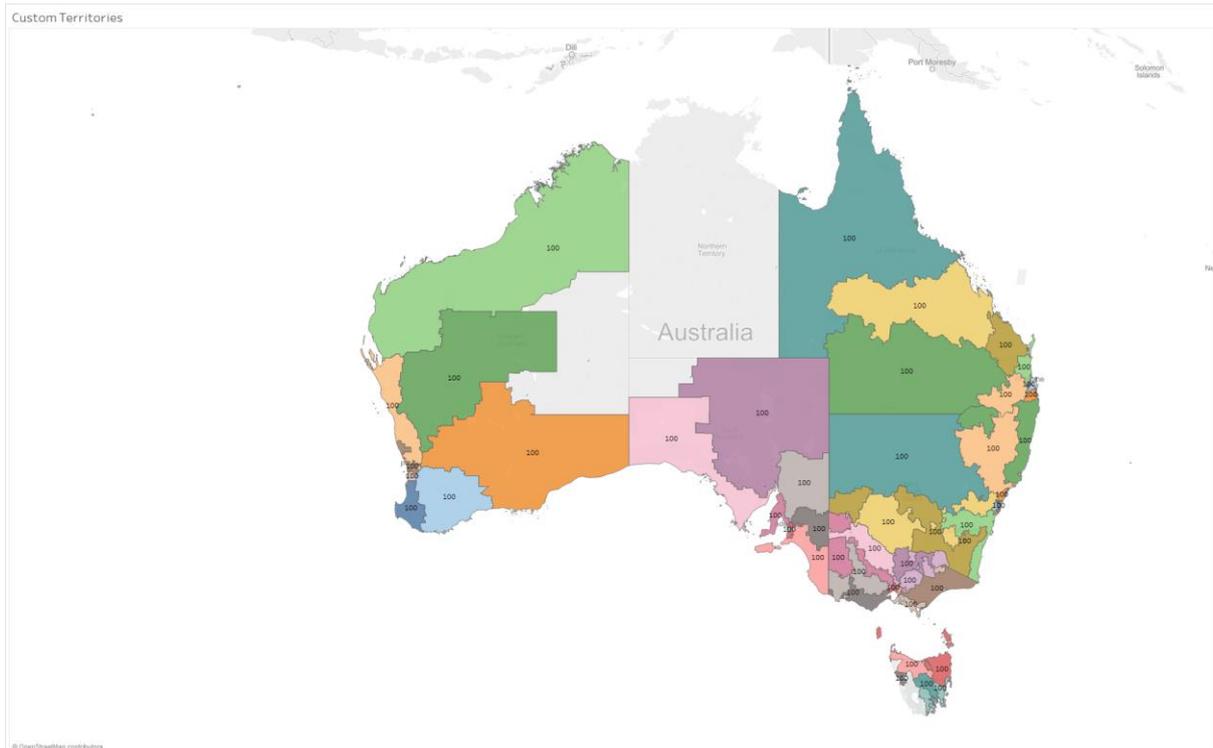


Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
AreaCode.tds	Tableau Datasource	2 KB	No	9 KB	82%	9/06/2016 1:29 PM
City.tds	Tableau Datasource	2 KB	No	13 KB	85%	9/06/2016 1:29 PM
CMSA.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Congress.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Country.tds	Tableau Datasource	3 KB	No	17 KB	87%	9/06/2016 1:29 PM
County.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
GEOCODING.FDB	FDB File	118,407 KB	No	359,280 KB	68%	9/06/2016 1:28 PM
State.tds	Tableau Datasource	2 KB	No	14 KB	86%	9/06/2016 1:29 PM
Suburb.tds	Tableau Datasource	3 KB	No	15 KB	86%	9/06/2016 1:29 PM
ZipCode.tds	Tableau Datasource	2 KB	No	8 KB	81%	9/06/2016 1:29 PM

Dadurch wird die daraus resultierende TWBX-Datei a) sehr groß, weil die Geokodierungs-Nutzlast komprimiert etwa 110 MB ausmacht, und b) beim Öffnen ziemlich langsam, weil die anfängliche Dekomprimierung mit einem viel größeren Datenbestand ausgeführt werden muss. Ein effizienterer Ansatz ist es, die Daten nicht als benutzerdefinierte Geokodierungsrolle zu importieren, sondern eher eine Verschmelzung in der Arbeitsmappe zu verwenden, um die Analysedaten mit den geografischen Daten zu kombinieren. Bei diesem Ansatz wird die Datei geocoding.fdb nicht eingebettet und die TWBX enthält nur die Analyse- und die benutzerdefinierten Geokodierungsdaten.

Benutzerdefinierte Gebiete

Eine neue Funktion von Tableau 10 sind die benutzerdefinierten Gebiete, mit denen Benutzer Bereiche aus internen Geokodierungsdatenbanken kombinieren können, um aggregierte Regionen zu erstellen.



Das anfängliche Rendern der benutzerdefinierten Gebiete, die auf zahlreichen Regionen auf niedriger Ebene basieren, kann sehr viel Zeit in Anspruch nehmen, verwenden Sie diese Funktion also mit Vorsicht. Wenn das abgeschlossen ist, werden die benutzerdefinierten Gebiete allerdings im Cache abgelegt. Es ist also eine gute Leistung zu erwarten.

Ausgefüllte Karten vs. Punktkarten

Ausgefüllte Kartenmarkierungen (in einer Karte oder als Markierung in einer anderen Art von Grafik verwendet) sind teuer, wenn clientseitiges Rendern verwendet wird. Die Polygondaten für die Form müssen übertragen werden und das kann recht komplex sein. Ziehen Sie stattdessen eine Symbolkarte in Betracht, wenn Sie beim Rendern eine geringe Leistung beobachten.

Polygonmarkierungen

Visualisierungen mit Polygonmarkierungen zwingen Tableau Server, [serverseitig zu rendern](#), was eine Auswirkung auf die Nutzererfahrung haben kann. Setzen Sie sie nur sparsam ein.

Weitere Faktoren

Große Kreuztabellen

In früheren Versionen dieses Dokuments haben wir empfohlen, große Kreuztabellen zu vermeiden, weil sie nur sehr langsam gerendert werden. Die zugrunde liegende Mechanik dieses Visualisierungstyps wurde in neueren Versionen verbessert, sodass Kreuztabellen jetzt ebenso schnell wie andere Rasterdiagramme gerendert werden. Wir empfehlen jedoch weiterhin, die Verwendung großer Kreuztabellen sorgfältig zu bedenken. Dafür müssen sehr viele Daten aus der zugrunde liegenden Datenquelle gelesen werden, die für den Analyse nicht nützlich sind.

QuickInfos

Wenn eine Dimension auf den QuickInfo-Container gezogen wird, wird sie mit Attributfunktion ATTR() aggregiert. Dafür müssen zwei Aggregationen in der zugrunde liegenden Datenquelle ausgeführt werden – MIN() und MAX() – und beide Ergebnisse werden an den Ergebnissatz

zurückgegeben. Weitere Informationen finden Sie im Abschnitt [ATTR\(\) verwenden](#) weiter unten im Dokument.

Wenn die Möglichkeit, mehrere Dimensionswerte zu sehen, irrelevant für Sie ist, können Sie die effizientere Lösung wählen, anstelle der Standardfunktion ATTR() nur eine Aggregation zu verwenden. Wählen Sie entweder MIN() oder MAX() – ganz gleich welche, aber wählen Sie eine und bleiben Sie dabei, um die Cache-Trefferquote zu maximieren.

Eine weitere Option: Wenn Sie wissen, dass es die visuelle Detailgenauigkeit Ihrer Visualisierung nicht beeinträchtigt, können Sie die Dimension auf den Detailgenauigkeits-Container ziehen, anstatt auf den QuickInfo-Container. Dies führt dazu, dass das Dimensionsfeld direkt in den SELECT- und GROUP BY-Klauseln der Abfrage verwendet wird. Testen Sie, ob es besser funktioniert als die einzelne Aggregation. Dies ist abhängig von der Leistung Ihrer Datenplattform.

Legenden

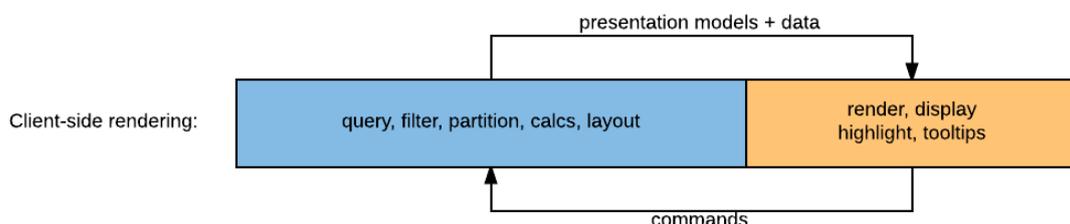
Legenden verursachen im Allgemeinen keine Leistungsprobleme, weil ihre Domänen aus dem Abfrageergebnis-Cache bedient werden. Sie können jedoch zum Rendering-Aufwand beitragen, wenn die enumerierte Domäne groß ist, weil die Daten zum Client-Browser übertragen werden müssen. Falls dies zutrifft, ist die die Legende im Allgemeinen nicht nützlich. Es ist also besser, sie zu entfernen.

Seitencontainer

Einige Benutzer denken, dass der Seitencontainer ebenso funktioniert wie der Filtercontainer, d. h., dass er die Anzahl der von der Datenquelle zurückgegebenen Datensätze reduziert. Das ist nicht richtig: Die Abfrage für das Arbeitsblatt gibt Datensätze für alle Markierungen von allen Seiten zurück. Wenn Sie eine Seitendimension mit einem hohen Maß an Kardinalität haben (d. h. viele Einzelwerte), kann dadurch die Größe des Arbeitsblatts signifikant ansteigen, was die Leistung beeinträchtigt. Verwenden Sie diese Funktion nur sparsam.

Clientseitiges vs. serverseitiges Rendern

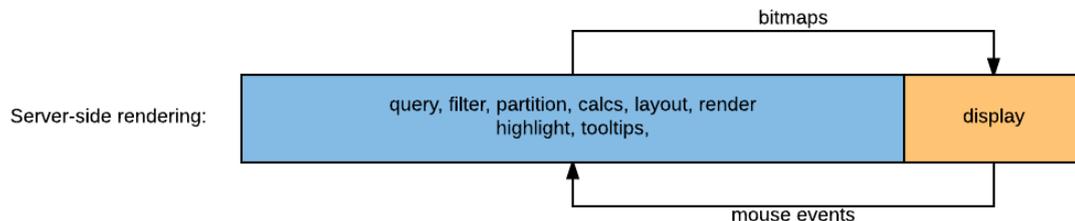
Bevor die Zellen und Daten der Ansicht in einem Client-Webbrowser angezeigt werden, werden sie abgerufen, interpretiert und gerendert. Tableau Server kann diesen Prozess im Client-Webbrowser oder auf dem Server ausführen. Das clientseitige Rendern ist der Standardmodus, weil die Handhabung des Renderns und aller Interaktionen auf dem Server einen verstärkten Netzwerkdatenverkehr und Umlaufzeitverzögerungen zur Folge haben kann. Beim clientseitigen Rendern laufen viele Interaktionen für Ansichten schneller ab, weil sie direkt vor Ort im Browser interpretiert und gerendert werden.



(Blau – auf dem Server; Orange – im Client-Browser)

Einige Ansichten werden allerdings auf dem Server effizienter gerendert, weil dort mehr Rechenleistung zur Verfügung steht. Serverseitiges Rendern ist bei einer Ansicht sinnvoll, die so komplex ist, dass die Bilddateien wesentlich weniger Bandbreite erfordern als die zum Erstellen der Bilder verwendeten Daten.

Und da Tablets normalerweise eine geringere Leistung haben als PCs, können sie komplexe Ansichten nicht so gut handhaben. Ein Nachteil ist allerdings, dass einfache Interaktionen, wie etwa QuickInfo und Markieren, beim serverseitigen Rendern langsam sein können, weil die Daten hin und her übertragen werden müssen.



(Blau – auf dem Server; Orange – im Client-Browser)

Tableau Server ist so konfiguriert, dass diese Situationen mithilfe eines Komplexitäts-Schwellenwerts als Auslöser für das Rendern einer Ansicht auf dem Server bzw. im Webbrowser automatisch gehandhabt werden. Der Schwellenwert ist für PCs und Mobilgeräte verschieden. Es gibt also Fälle, wo eine von einem PC-Webbrowser geöffnete Ansicht clientseitig gerendert wird, jedoch die gleiche, von einem Tablett-Webbrowser geöffnete Ansicht serverseitig gerendert wird. Das Filtern kann auch das Verhalten beim Rendern verändern: Eine Arbeitsmappe wird möglicherweise ursprünglich beim Öffnen serverseitig gerendert, jedoch zum clientseitigen Rendern wechseln, wenn ein Filter angewendet wird. Wenn eine Visualisierung Polygonmarkierungen oder den Seitencontainer verwendet, wird nur serverseitig gerendert. Dies trifft auch zu, wenn sie ansonsten die Kriterien für clientseitiges Rendern erfüllt. Verwenden Sie diese Funktionen also mit Vorsicht.

Als Administrator können Sie diese Einstellung für PCs und Tablets testen oder feinabstimmen. Detailliertere Hinweise finden Sie unter dem folgenden Link:

https://onlinehelp.tableau.com/current/server/de-de/browser_rendering.htm

Effiziente Filter

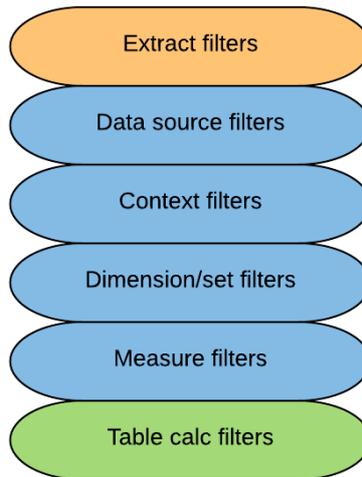
Filtern ist eine besonders leistungs- und ausdrucksstarke Funktion in Tableau. Ineffiziente Filter sind jedoch die häufigsten Gründe für schlechte Leistungen von Arbeitsmappen und Dashboards. In den folgenden Abschnitten werden mehrere bewährte Verfahren für die Arbeit mit Filtern beschrieben.

Hinweis: Filter werden durch Indizes in der Datenquelle deutlich effizienter. Genauere Hinweise finden Sie im Abschnitt über Indizes.

Filtertypen

Filterpriorität

Filter werden in Tableau wie folgt angewendet:



Extraktfilter

Diese Filter werden nur bei Datenextrakten eingesetzt, werden in diesem Fall jedoch logisch, vor allen anderen Filtern angewendet. Sie begrenzen die von der zugrunde liegenden Datenquelle abgerufenen Daten und können entweder Dimensions- oder Kennzahlenfilter sein. Zusätzlich können Sie entweder TOP oder SAMPLE ausführen, um die Anzahl der zurückgegebenen Datensätze zu reduzieren, je nach der verwendeten Quelldatenplattform.

Datenquellenfilter

Datenquellenfilter sind die höchste Filterebene, die bei Direktverbindungen verfügbar ist. Ein wichtiger Unterschied zwischen Datenquellenfiltern und Kontextfiltern ist, dass Datenquellenfilter die gesamte Datenquelle umschließen, wohingegen Kontextfilter für jedes einzelne Arbeitsblatt festgelegt werden. Das bedeutet Folgendes: Wenn Datenquellenfilter in einer veröffentlichten Datenquelle eingesetzt werden, können Sie erzwungen werden, während Kontextfilter auf Arbeitsblattebene angewendet werden.

Datenquellenfilter können eine effektive Methode sein, eine Einschränkung auf eine Datenquelle anzuwenden, damit Endbenutzer nicht versehentlich eine Massenabfrage ausführen. Zum Beispiel könnten Sie einen Datenquellenfilter einrichten, der Abfragen in einer Transaktionstabelle auf die letzten sechs Monate beschränkt.

Kontextfilter

Standardmäßig werden alle Filter, die Sie in Tableau festlegen, unabhängig voneinander berechnet. Jeder Filter greift also auf alle Zeilen der Datenquelle zu, ohne andere Filter zu berücksichtigen. Durch die Angabe eines Kontextfilters können Sie alle anderen von Ihnen definierten Filter als abhängig definieren, weil sie nur die Daten verarbeiten, die der Kontextfilter durchlässt.

Sie sollten Kontextfilter verwenden, wenn sie erforderlich sind, um die richtige Antwort zu erhalten (z. B. gefilterte Oberste N). Stellen Sie sich zum Beispiel eine Ansicht vor, in der die obersten 10 Produkte nach SUM(Umsatz), gefiltert nach Regionen angezeigt werden. Ohne einen Kontextfilter wird die folgende Abfrage ausgeführt:

```
SELECT [Superstore APAC].[Product Name] AS [Product Name],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
     INNER JOIN (
       SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
                  SUM([Superstore APAC].[Sales]) AS [$_alias_0]
       FROM [dbo].[Superstore APAC] [Superstore APAC]
```

```

GROUP BY [Superstore APAC].[Product Name]
ORDER BY 2 DESC
) [t0] ON ([Superstore APAC].[Product Name] = [t0].[Product Name])
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Product Name]

```

Dabei wird der Beitrag in Ozeanien zu den obersten 10 Produkten zurückgegeben. Wenn Sie jedoch die obersten 10 Produkte in der Region Ozeanien ermitteln wollten, würden Sie den Regionfilter zum Kontext hinzufügen und die folgende Abfrage würde ausgeführt werden:

```

SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok],
SUM([Superstore APAC].[Sales]) AS [$__alias__0]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Product Name]
ORDER BY 3 DESC

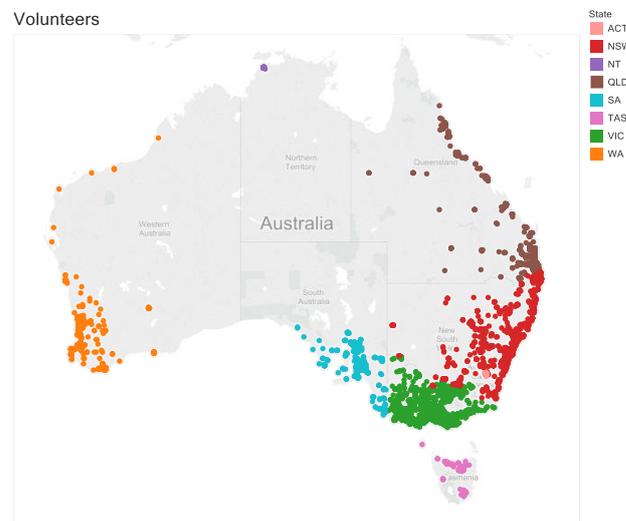
```

Ursprünglich wurden Kontextfilter als temporäre Tabelle in die Datenquelle implementiert. Das ist jetzt nicht mehr der Fall. In fast allen Situationen werden Kontextfilter als Teil der Datenquellenabfrage implementiert (siehe oben) oder lokal in der Datenengine verarbeitet.

Sie sollten Kontextfilter nicht mehr als Mechanismus einsetzen, um die Abfrageleistung zu verbessern.

Filtern von Kategoriedimensionen

Denken Sie an die folgende Visualisierung einer Karte von Australien mit Markierungen nach Postleitzahl:



Es gibt mehrere Möglichkeiten, wie Sie die Karte filtern können, um nur die Postleitzahlen für Westaustralien (orangefarbene Punkte) zu zeigen:

- Sie können alle Markierungen in WA auswählen und nur die Auswahl beibehalten.
- Sie können alle Markierungen außerhalb von WA auswählen und die Auswahl ausschließen.
- Sie können nur ein anderes Attribut, beispielsweise die Bundesstaat-Dimension beibehalten.
- Sie können nach Bereichen filtern, entweder nach Postleitzahlwerten oder Werten über Längen- und Breitengrade.

Diskret

Bei den ersten beiden Optionen stellen Sie möglicherweise fest, dass die Filter „Nur beibehalten“ oder „Ausschließen“ keine gute Leistung bewirken. In der Tat sind sie oft langsamer als bei ungefilterten Datensätzen. Der Grund dafür ist, dass sie als diskrete Liste von Postleitzahlwerten ausgedrückt werden, die von der DBMS gefiltert werden. Entweder durch eine WHERE IN-Klausel oder, falls viele Werte vorhanden sind, indem eine temporäre Tabelle mit den ausgewählten Werten angelegt und ein INNER JOIN mit dieser und der Haupttabelle / den Haupttabellen verwendet wird. Bei einem umfangreichen Satz von Markierungen kann dies zu einer sehr teuren Abfrage werden.

Die dritte Option ist in diesem Beispiel schneller, weil der dabei entstehende Filter (`WHERE STATE= "Western Australia"`) sehr einfach ist und von der Datenbank einfach verarbeitet werden kann. Dieser Ansatz verliert aber an Wirksamkeit, wenn die Anzahl der Dimensionselemente, die für den Ausdruck des Filters benötigt werden, zunimmt. Sie erreicht schließlich die Leistung des Lassos und der „Nur beibehalten“-Option.

Wertebereich

Der Ansatz mit Wertebereichsfilter ermöglicht es der Datenbank ebenfalls, eine einfache Filterklausel zu evaluieren (entweder `WHERE POSTCODE >= 6000 AND POSTCODE <= 7000` oder `WHERE LONGITUDE < 129`), was eine schnelle Ausführung zur Folge hat. Dieser Ansatz wird jedoch im Gegensatz zu einem Filter für eine verbundene Dimension mit erhöhter Kardinalität der Dimensionen nicht komplexer.

Das Ergebnis: Wertebereichsfilter sind oft schneller in der Auswertung als lange Listen mit diskreten Werten und sollten bevorzugt anstelle von „Nur beibehalten“- oder „Ausschließen“-Filtern für umfangreiche Markierungssätze verwendet werden.

Aufteilungsfilter

Aufteilungsfilter sind Filter für Dimensionen, die nicht in der Visualisierung verwendet werden (d. h. sie sind nicht Teil der Visualisierungs-Detailgenauigkeit). Sie haben beispielsweise eine Visualisierung mit Gesamtumsätzen pro Land, aber die Visualisierung ist nach Regionen gefiltert. In diesem Beispiel ist der Abfrageablauf wie folgt:

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
WHERE ([Superstore APAC].[Region] = 'Oceania')
GROUP BY [Superstore APAC].[Country]
```

Diese Filter werden komplexer, wenn wir nach dem Ergebnis einer Aggregation aufteilen. Wenn wir die obige Visualisierung nicht nach Region filtern, sondern den Umsatz für die 10 Produkte mit den höchsten Verkaufszahlen anzeigen, muss Tableau zwei Abfragen ausführen: eine auf Produktebene, um die 10 Produkte mit den höchsten Verkaufszahlen zu isolieren, und dann eine andere auf Landesebene, eingeschränkt durch die Ergebnisse der ersten Abfrage:

```
SELECT [Superstore APAC].[Country] AS [Country],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
INNER JOIN (
  SELECT TOP 10 [Superstore APAC].[Product Name] AS [Product Name],
               SUM([Superstore APAC].[Profit]) AS [$_alias_0]
  FROM [dbo].[Superstore APAC] [Superstore APAC]
  GROUP BY [Superstore APAC].[Product Name]
  ORDER BY 2 DESC
) [t0] ON ([Superstore APAC].[Product Name] = [t0].[Product Name])
GROUP BY [Superstore APAC].[Country]
```

Seien Sie vorsichtig, wenn Sie Aufteilungsfiler verwenden, weil ihre Evaluierung möglicherweise teuer sein kann. Beachten Sie auch Folgendes: Da die Dimensionen nicht im Abfrageergebnis-Cache vorliegen, können wir bei Aufteilungsfilern keine schnelle Filterung im Browser ausführen (siehe [Abschnitt weiter oben](#) über clientseitiges vs. serverseitiges Rendern).

Datenquellenübergreifende Filter

Das datenquellenübergreifende Filtern ist eine neue Funktion in Tableau 10. So kann ein Filter auf mehrere Datenquellen angewendet werden, die ein oder mehrere gemeinsame Felder haben. Beziehungen werden genau wie beim Verschmelzen definiert: automatisch auf der Grundlage von Name-/Typ-Übereinstimmungen oder manuell über benutzerdefinierte Beziehungen im Datenmenü.

Datenbankübergreifende Filter weisen die gleichen Leistungsimplikationen auf wie Schnellfilter in einem Dashboard. Wenn Sie sie verändern, werden möglicherweise mehrere Bereiche aktualisiert und mehrere Abfragen sind erforderlich. Verwenden Sie sie sparsam und falls Sie erwarten, dass die Benutzer mehrere Änderungen vornehmen, blenden Sie die Schaltfläche "Übernehmen" ein, damit Abfragen erst ausgelöst werden, wenn die Auswahl abgeschlossen ist.

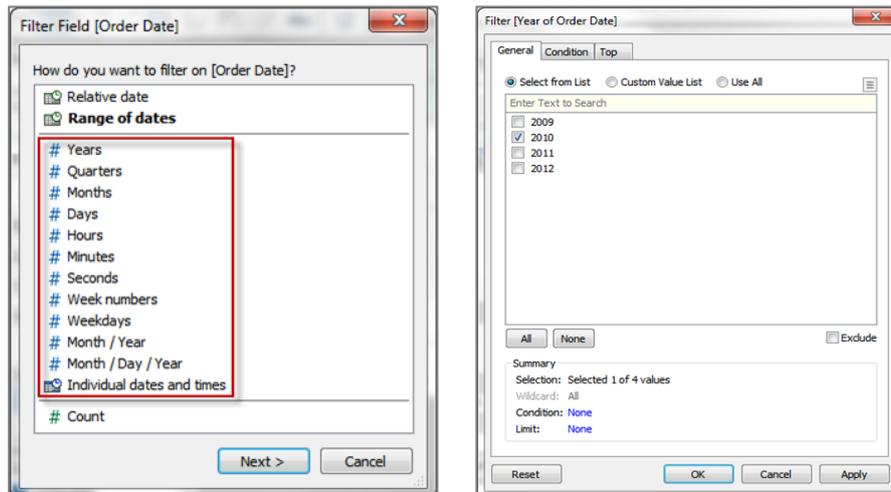
Beachten Sie außerdem, dass die Filterdomäne von der „primären“ Datenquelle übernommen wird, der zuerst benutzten Datenquelle des Blatts, in dem der Filter eingerichtet wurde. Wenn ein verbundenes Feld über verschiedene Domänen in unterschiedlichen Datenquellen verfügt, müssen Sie sorgfältig prüfen, welche Sie verwenden, denn ein und derselbe Filter kann unterschiedliche Werte anzeigen, wie unten gezeigt:



Filtern von Datumswerten: diskret, Bereich, relativ

Datumfelder stellen eine besondere Art von Dimension dar, die in Tableau häufig anders als standardmäßige Kategoriedaten behandelt werden. Dies gilt vor allem, wenn Sie Datumfilter erstellen. Datumfilter werden sehr häufig verwendet und fallen in drei Kategorien: Filter vom Typ „Relatives Datum“, bei denen ein Datumsbereich angezeigt wird, der relativ zu einem bestimmten Tag gilt. Filter vom Typ „Datumsbereich“, bei denen ein definierter Bereich diskreter Datumsangaben angezeigt wird. Datumfilter vom Typ „Diskret“, bei denen einzelne Datumsangaben angezeigt werden, die Sie aus einer Liste ausgewählt haben. Wie im Abschnitt oben gezeigt wird, können die benutzten Methoden wesentliche Auswirkungen auf die Effizienz der sich daraus ergebenden Abfrage haben.

Diskret



Es kann vorkommen, dass Sie beim Filtern bestimmte einzelne Datumsangaben oder ganze Datumsebenen einbeziehen möchten. Diese Art von Filter wird als diskreter Datumfilter bezeichnet, weil Sie anstelle eines Bereichs diskrete Werte definieren. Dieser Filter führt dazu, dass der Datumsausdruck als dynamische Berechnung an die Datenbank übermittelt wird:

```
SELECT [FactSales].[Order Date], SUM([FactSales].[SalesAmount])
FROM [dbo].[FactSales] [FactSales]
WHERE (DATEPART(year,[FactSales].[Order Date])) = 2010)
GROUP BY [FactSales].[Order Date]
```

In den meisten Fällen bewertet die Intelligenz der Abfrageoptimierer die DATEPART-Berechnung, aber es gibt Szenarios, bei denen der Einsatz von diskreten Datenfiltern zu einer schlechten Abfrageleistung führt. Dies ist beispielsweise der Fall, wenn Sie eine partitionierte Tabelle mit einem diskreten Datenfilter für den Datums-Partitionsschlüssel abfragen. Da die Tabelle nicht für den DATEPART-Wert partitioniert ist, werden einige Datenbanken die Berechnung über alle Partitionen auswerten, um Datensätze zu finden, die mit den Kriterien übereinstimmen, auch wenn dies gar nicht nötig ist. In diesem Fall wird sich die Leistung möglicherweise wesentlich verbessern, wenn Sie einen Datumsbereichsfilter oder einen relativen Filter mit einem bestimmten Ankerdatum verwenden.

Eine Methode zur Optimierung der Leistung bei dieser Art von Filtern besteht in der Materialisierung der Berechnung mithilfe eines Datenextrakts. Zuerst erstellen Sie ein berechnetes Feld, das die DATEPART-Funktion explizit implementiert. Wenn Sie dann einen Tableau-Datenextrakt erstellen, wird dieses berechnete Feld als gespeicherte Werte im Extrakt materialisiert (weil die Ausgabe des Ausdrucks deterministisch ist). Das Filtern des berechneten Felds anstelle des dynamischen Ausdrucks ist schneller, weil der Wert einfach gesucht werden kann und nicht zum Abfragezeitpunkt berechnet werden muss.

Datumsbereich

Diese Art von Filter wird benutzt, wenn Sie einen Bereich zusammenhängender Datumswerte angeben möchten. Dies bewirkt, dass die folgende Abfragestruktur an die Datenbank übermittelt wird:

```
SELECT SUM([factOrders].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[factOrders] [factOrders]
WHERE (([factOrders].[Order Date] >= {d '2015-01-01'}) AND
([factOrders].[Order Date] <= {d '2016-12-31'}))
GROUP BY ()
```

Diese Art der WHERE-Klausel ist sehr effizient für Abfrageoptimierer und gestattet, Ausführungspläne, Indizes und Partitionen in vollem Umfang auszunutzen. Wenn Sie nach dem Hinzufügen von diskreten Datumfiltern sehr langsame Abfragen feststellen, denken Sie daran, sie über Datenfilter durch Bereiche zu ersetzen und zu prüfen, ob sich daraus ein Unterschied ergibt.

Relativ

Mit einem Filter vom Typ „Relatives Datum“ können Sie einen Datumsbereich definieren, der in Abhängigkeit davon aktualisiert wird, an welchem Datum und zu welcher Uhrzeit Sie die Ansicht öffnen. Angenommen, Sie möchten den Umsatz im bisherigen Jahr, alle Datensätze der letzten 30 Tage oder die in der letzten Woche behobenen Fehler anzeigen. Filter vom Typ „Relatives Datum“ können auch relativ zu einem bestimmten Ankerdatum verwendet werden, nicht nur für den jeweiligen aktuellen Tag.

```
SELECT SUM([factOrders].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[factOrders] [factOrders]
WHERE (([factOrders].[Order Date] >= {ts '2015-01-01 00:00:00'}) AND
([factOrders].[Order Date] < {ts '2017-01-01 00:00:00'}))
GROUP BY ()
```

Wie Sie sehen, benutzt die sich daraus ergebende WHERE-Klausel eine Datumsbereichssyntax. Dies ist daher ebenfalls eine effiziente Form eines Datumsfilters.

Beachten Sie Folgendes: Aufgrund der sich ändernden Datumsfilter, die sich auf das aktuelle Datum / die aktuelle Uhrzeit beziehen – entweder über die Option „relativer Datumsfilter“ oben oder über eine explizite Verwendung von NOW() oder TODAY() in einer Filterformel – ist der Abfrageergebnis-Cache nicht sehr effizient für Abfragen, die sie benutzen. Tableau markiert sie im Cache-Server als „vergängliche Abfragen“ und ihre Ergebnisse werden nicht so lange gespeichert wie andere Abfrageergebnisse.

Filterkarten

Wenn zu viele Filterkarten angezeigt werden, wird die Verarbeitung verlangsamt, insbesondere, wenn Sie sie auf „Nur relevante Werte“ einstellen und viele diskrete Listen vorhanden sind. Versuchen Sie es mit einem Ansatz, der auf Guided Analytics beruht und verwenden Sie Aktionsfilter in einem Dashboard. Wenn Sie eine Ansicht mit vielen Filtern erstellen, um sie höchst interaktiv zu machen, sollten Sie sich die Frage stellen, ob mehrere Dashboards mit verschiedenen Ebenen nicht besser funktionieren würden (Antwort: ja, höchstwahrscheinlich).

Enumeriert vs. nicht enumeriert

Enumerierte Filterkarten veranlassen Tableau, eine Abfrage in der Datenquelle für Feldwerte vorzunehmen, bevor die Filterkarte gerendert werden kann. Dazu gehören:

- Mehrere Werte (Liste) – *alle Dimensionselemente*
- Einzelwert (Liste) – *alle Dimensionselemente*
- Kompaktiliste – *alle Dimensionselemente*
- Schieberegler – *alle Dimensionselemente*
- Kennzahlfilter – *MIN- und MAX-Werte*
- Datumsbereichsfilter – *MIN- und MAX-Werte*

Nicht enumerierte Filterkarten benötigen hingegen kein Wissen über potenzielle Feldwerte. Dazu gehören:

- Liste mit benutzerdefinierten Werten
- Platzhalterübereinstimmung
- Filter vom Typ „Relatives Datum“
- Datumsfilter zum Durchsuchen eines Zeitraums

Nicht enumerierte Filterkarten reduzieren daher die Anzahl von Abfragen im Zusammenhang mit Filtern, die von der Datenquelle ausgeführt werden müssen. Außerdem rendern nicht enumerierte Filterkarten schneller, wenn viele Dimensionselemente anzuzeigen sind.

Die Verwendung von nicht enumerierten Filterkarten kann die Leistung verbessern, aber dies geht auf Kosten des visuellen Kontexts für den Anwender.

Relevante Werte

Enumerierte Filterkarten können so eingestellt werden, dass sie die potenziellen Feldwerte auf drei Arten anzeigen:

- Alle Werte in der Datenbank – Wenn Sie diese Option auswählen, werden unabhängig von den anderen Filtern der Ansicht alle Werte der Datenbank angezeigt. Der Filter muss die Datenbank nicht erneut abfragen, wenn andere Filter geändert werden.

- Alle Werte im Kontext – Diese Option ist nur verfügbar, wenn Sie aktive Kontextfilter haben. Die Filterkarte zeigt alle Werte im Kontext an, unabhängig von anderen Filtern in der Ansicht. Der Filter muss die Datenbank nicht erneut abfragen, wenn Dimensions- oder Kennzahlfilter geändert werden. Wenn sich jedoch der Kontext ändert, muss erneut abgefragt werden.
- Nur relevante Werte – Wenn Sie diese Option auswählen, werden auch andere Filter berücksichtigt, und es werden nur Werte angezeigt, die die Bedingungen dieser Filter erfüllen. Ein Filter für Bundesstaaten kann z. B. nur die Staaten für „East“ anzeigen, wenn ein Filter für „Region“ entsprechend eingerichtet wurde. Der Filter muss daher die Datenbank erneut abfragen, wenn andere Filter geändert werden.

Sie sehen: Die Einstellung „Nur relevante Filter“ kann hilfreich sein, damit der Benutzer relevante Auswahlen treffen kann. Sie erhöht aber die Anzahl der Abfragen, die ausgeführt werden müssen, während die Benutzer mit dem Dashboard arbeiten. Daher sollte sie nur eingeschränkt verwendet werden.

Alternativen für Filterkarten

Es gibt Alternativen zur Verwendung von Filterkarten, die ein ähnliches analytisches Ergebnis liefern, jedoch ohne den zusätzlichen Abfrageaufwand. Sie könnten zum Beispiel einen Parameter und Filter auf der Grundlage der Benutzerauswahl erstellen.

- VORTEILE:
 - Parameter benötigen vor dem Rendern keine Datenbankabfrage.
 - Parameter + berechnete Felder können Logik implementieren, die komplexer ist als bei einem einfachen Feldfilter möglich ist.
 - Parameter können verwendet werden, um über mehrere Datenquellen zu filtern. Bei Versionen vor Tableau 10 arbeiten die Filter hingegen nur mit einer einzelnen Datenquelle. In Tableau 10 und höher können Filter dafür eingerichtet werden, datenquellenübergreifend zu arbeiten.
- NACHTEILE:
 - Parameter haben nur einen einzelnen Wert. Sie können sie nicht verwenden, wenn der Benutzer mehrere Werte auswählen soll.
 - Parameter sind nicht dynamisch. Die Liste der Werte wird beim Erstellen definiert. Sie werden nicht auf der Grundlage der Werte im DBMS aktualisiert.

Eine andere Alternative ist die Verwendung von Filteraktionen zwischen Ansichten.

- VORTEILE:
 - Aktionen unterstützen die Auswahl mehrerer Werte durch visuelles Lasso oder Klicken mit Strg/Umschalttaste.
 - Aktionen zeigen eine dynamische Liste von Werten, die zur Laufzeit ausgewertet wird.
 - Aktionen können verwendet werden, um über mehrere Datenquellen zu filtern. Bei Versionen vor Tableau 10 arbeiten Filter hingegen nur mit einer einzelnen Datenquelle. In Tableau 10 und höher können Filter dafür eingerichtet werden, datenquellenübergreifend zu arbeiten.
- NACHTEILE:
 - Das Einrichten von Filteraktionen ist komplexer als bei Filterkarten.

- Aktionen haben nicht die gleiche Benutzerschnittstelle wie Parameter oder Filterkarten. Im Allgemeinen benötigen sie mehr Bildschirmfläche für die Anzeige.
- Die Aktionsquelle muss weiterhin die Datenquelle abfragen, profitiert aber vom Zwischenspeicher in der Tableau-Verarbeitungspipeline.

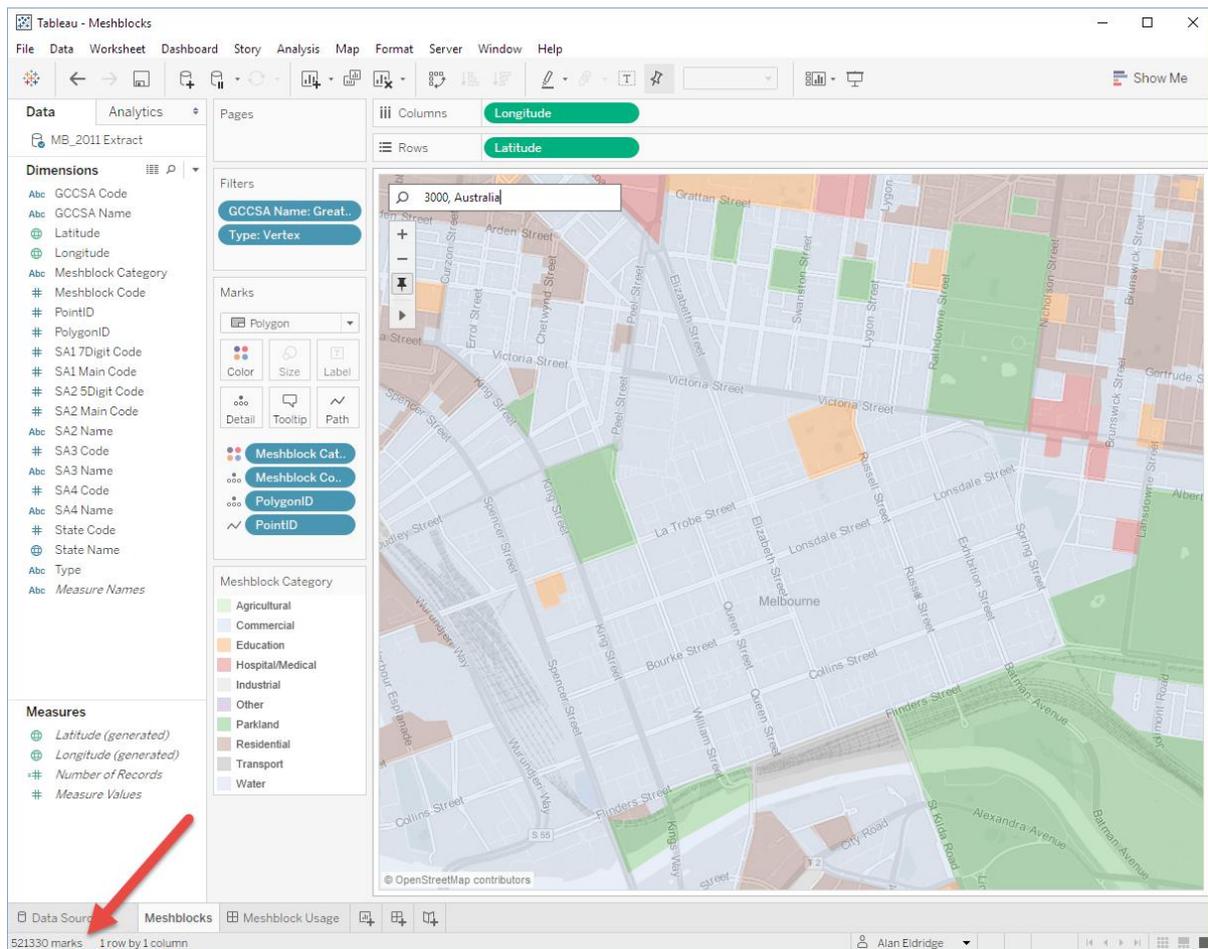
Eine weiterführende Diskussion über alternative Designtechniken, die weniger auf Filterkarten basieren, finden Sie [im Abschnitt weiter oben](#) über gutes Dashboard-Design.

Benutzerfilter

Wenn eine Arbeitsmappe Benutzerfilter enthält – entweder über das Dialogfeld „Benutzerfilter erstellen“ oder durch berechnete Felder, die eine der integrierten Benutzerfunktionen wie ISMEMBEROF() enthält – wird der Modell-Cache nicht in mehreren Benutzersitzungen genutzt. Dies kann eine wesentlich niedrigere Cache-Trefferquote zur Folge haben, was wiederum mehr Arbeit für Tableau Server bedeutet. Verwenden Sie diese Filtertypen mit Bedacht.

Anpassen der Größe vs. Filtern

Wenn Sie eine Visualisierung mit einer großen Anzahl von Markierungen vergrößern, werden die nicht sichtbaren Markierungen nicht herausgefiltert. Es wird lediglich das Sichtfenster auf die Daten verändert. Die Gesamtzahl der zu verarbeitenden Markierungen verändert sich nicht, wie Sie am folgenden Bild erkennen können.



Wenn Sie nur eine Teilmenge der Daten benötigen, filtern Sie die nicht benötigten Daten heraus und lassen Sie die Auto-Zoom-Funktion von Tableau das Sichtfenster festlegen.

Liegt es an meinen Berechnungen?

In vielen Fällen stellen Ihre Quelldaten nicht alle Felder bereit, die Sie benötigen, um alle Ihre Fragen zu beantworten. Berechnete Felder helfen Ihnen dabei, alle Dimensionen und Kennzahlen für Ihre Analyse zu erstellen.

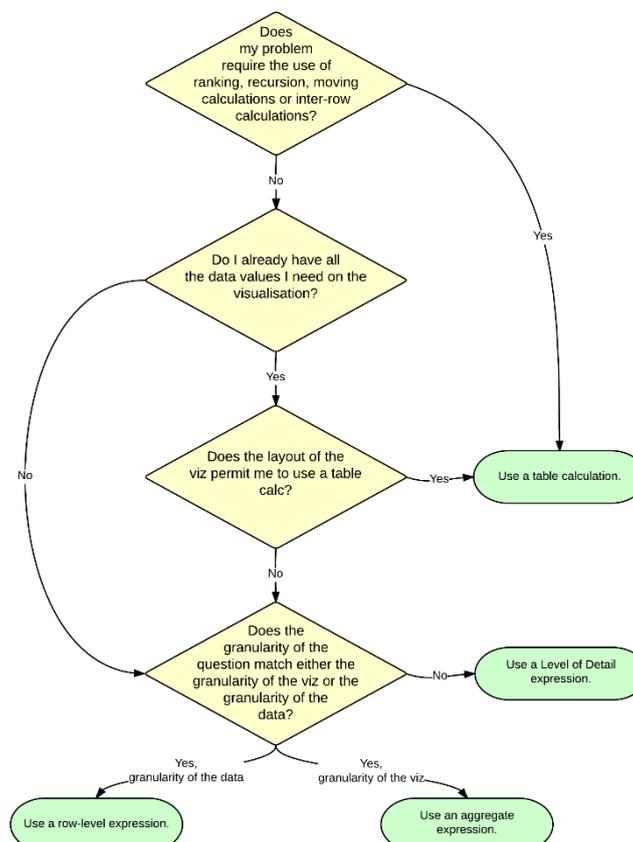
Innerhalb eines berechneten Feldes können Sie eine hartkodierte Konstante (beispielsweise einen Steuersatz) definieren, einfache mathematische Berechnungen wie Subtraktion oder Addition (beispielsweise Umsatz minus Kosten) anstellen, komplexere mathematische Formeln nutzen, logische Tests durchführen (beispielsweise IF/THEN, CASE), Datentypen konvertieren und mehr.

Nachdem ein berechnetes Feld definiert wurde, steht es in der gesamten Arbeitsmappe zur Verfügung, solange die Arbeitsblätter dieselbe Datenquelle verwenden. Sie können berechnete Felder in Ihrer Arbeitsmappe genau so verwenden wie Sie Dimensionen und Kennzahlen aus Ihren Quelldaten verwenden.

Es gibt vier verschiedene Berechnungstypen in Tableau:

- Berechnungen auf Zeilenebene
- Aggregierte Berechnungen
- Tabellenberechnungen
- LOD-Ausdrücke

Verwenden Sie das folgende Flussdiagramm als Leitfaden für die Wahl des besten Ansatzes:



Eine großartige Referenz für die Möglichkeiten bei der Ausführung komplexer Berechnungen sowie ein Forum, in dem Benutzer Lösungen für allgemeine Probleme finden, ist die Tableau-Berechnungsreferenzbibliothek:

<http://tabsoft.co/1IOsWz>

Berechnungstypen

Berechnungen auf Zeilenebene und aggregierte Berechnungen

Berechnungen auf Zeilenebene und aggregierte Berechnungen werden als Teil der an die Datenquelle gesendeten Abfrage ausgedrückt und deshalb von der Datenbank berechnet. Eine Visualisierung, in der der Gesamtwert des Umsatzes pro Jahr aus den Auftragsdaten gezeigt wird, würde beispielsweise die folgende Abfrage an die Datenquelle senden:

```
SELECT DATEPART(year,[OrdersFact].[Order Date]) AS [yr:Order Date:ok],
SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY DATEPART(year,[OrdersFact].[Order Date])
```

YEAR ist eine Berechnung auf Zeilenebene und **SUM(UMSATZ)** ist eine aggregierte Berechnung.

Im Allgemeinen lassen sich Berechnungen auf Zeilenebene und aggregierte Berechnungen sehr gut skalieren. Es gibt zahlreiche Tuning-Techniken für Datenbanken, die zur Leistungssteigerung angewendet werden können.

Beachten Sie, dass in Tableau die tatsächliche Datenbankabfrage nicht unbedingt eine direkte Übersetzung der grundlegenden Berechnungen sein muss, die in der Visualisierung verwendet werden. Die folgende Abfrage wird zum Beispiel ausgeführt, wenn die Visualisierung ein berechnetes Feld „Gewinnverhältnis“ enthält, definiert als $SUM([Profit])/SUM([Sales])$:

```
SELECT DATEPART(year,[OrdersFact].[Order Date]) AS [yr:Order Date:ok],
SUM([OrdersFact].[Profit]) AS
[TEMP(Calculation_0260604221950559)(1796823176)(0)],
SUM([OrdersFact].[Sales]) AS
[TEMP(Calculation_0260604221950559)(3018240649)(0)]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY DATEPART(year,[OrdersFact].[Order Date])
```

Tableau ruft die Elemente der Berechnung ab und führt die Divisionsfunktion auf der Clientebene aus. So wird gewährleistet, dass $SUM([Profit])$ und $SUM([Sales])$ auf der Jahresebene im Cache abgelegt werden und anderweitig in der Arbeitsmappe verwendet werden können, ohne auf die Datenquelle zugreifen zu müssen.

Schließlich kann die Abfragenverschmelzung (mehrere logische Abfragen in einer tatsächlichen Abfrage kombinieren) die in der Datenquelle ausgeführte Abfrage ändern. Dies kann in mehreren Kennzahlen aus mehreren Arbeitsblättern resultieren, die zu einer einzelnen Abfrage kombiniert werden, sofern die anderen Arbeitsblätter die gleiche Granularität aufweisen.

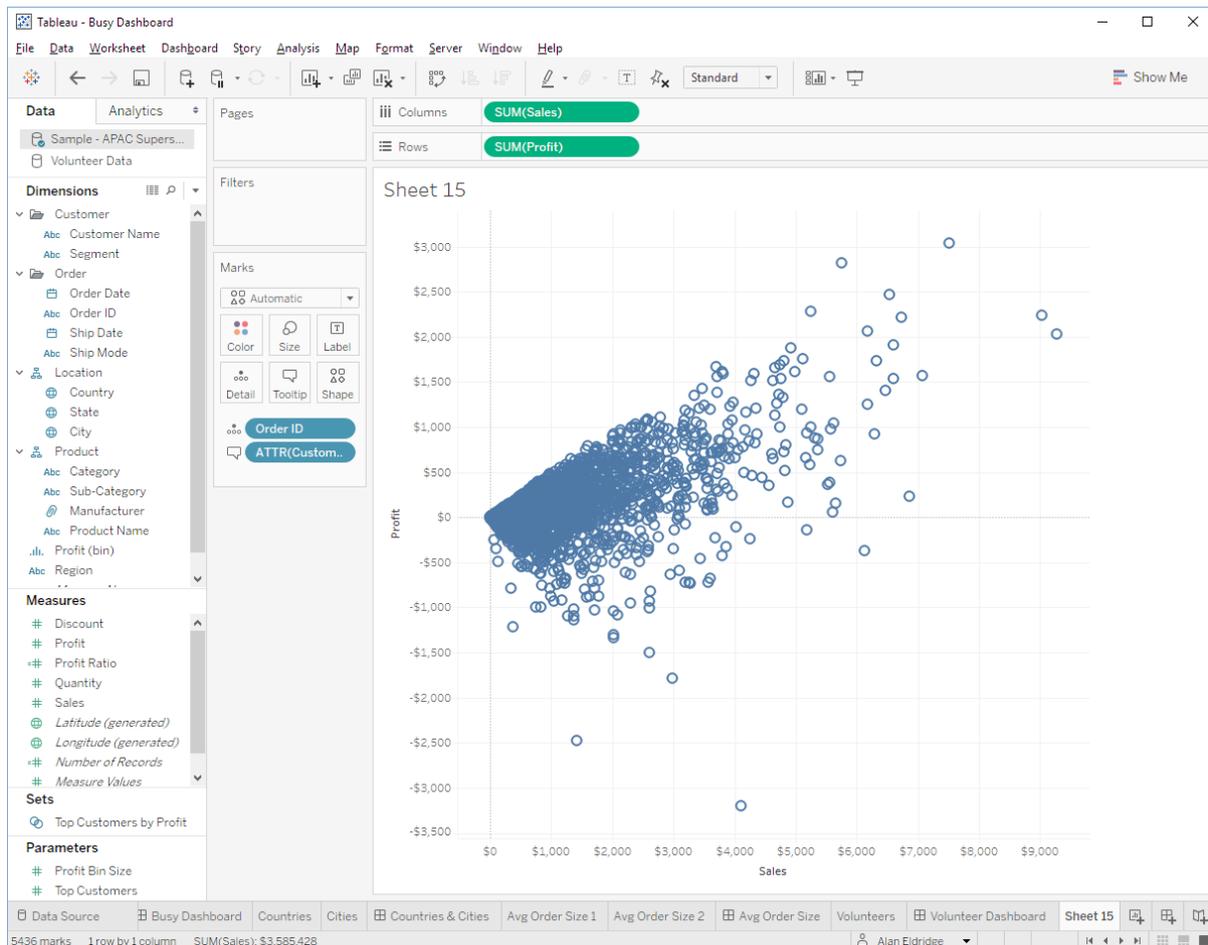
ATTR() verwenden

ATTR() ist eine leistungsstarke Funktion, die häufig als Aggregation für kategorische Dimensionen verwendet wird. Einfach ausgedrückt gibt sie einen Wert zurück, falls er eindeutig ist, ansonsten gibt sie * zurück. Technisch gesehen führt diese Funktion die folgenden logischen Tests durch:

```
IF MIN([dimension]) = MAX([dimension])
THEN [dimension]
ELSE "*"
END
```

Dafür müssen zwei Aggregationen in der zugrunde liegenden Datenquelle ausgeführt werden – MIN() und MAX() – und beide Ergebnisse werden an den Ergebnissatz zurückgegeben. Wenn die Möglichkeit, mehrere Dimensionswerte zu sehen, irrelevant für Sie ist, können Sie die effizientere Lösung wählen, nämlich nur eine Aggregation zu verwenden – entweder MIN() oder MAX(). Es ist unerheblich, welche Sie wählen. Wählen Sie jedoch eine und bleiben Sie dabei, um die Cache-Trefferquote zu maximieren.

Um zu zeigen, welche Auswirkungen ATTR() auf die Datenquelle hat, haben wir die folgende Visualisierung erstellt:



Wenn eine Dimension auf den QuickInfo-Container gezogen wird, wird sie mit der Attributfunktion ATTR() aggregiert. Daraus ergibt sich die folgende Abfrage:

```
SELECT [Superstore APAC].[Order ID] AS [Order ID],
       MAX([Superstore APAC].[Customer Name]) AS [TEMP(attr:Customer
Name:nk) (2542222306) (0)],
       MIN([Superstore APAC].[Customer Name]) AS [TEMP(attr:Customer
Name:nk) (3251312272) (0)],
       SUM([Superstore APAC].[Profit]) AS [sum:Profit:ok],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Order ID]
```

Wenn wir wissen, dass es für jede Auftrags-ID nur einen Kunden gibt, können wir die Aggregation in MIN() ändern und die Abfrage vereinfachen:

```

SELECT [Superstore APAC].[Order ID] AS [Order ID],
       MIN([Superstore APAC].[Customer Name]) AS [min:Customer Name:nk],
       SUM([Superstore APAC].[Profit]) AS [sum:Profit:ok],
       SUM([Superstore APAC].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[Superstore APAC] [Superstore APAC]
GROUP BY [Superstore APAC].[Order ID]

```

Aus der folgenden Tabelle ist die unterschiedliche Leistung der beiden Optionen ersichtlich (die Ergebnisse sind die Sekunden, die die Abfrage benötigt, und die Werte stammen aus der Protokolldateien):

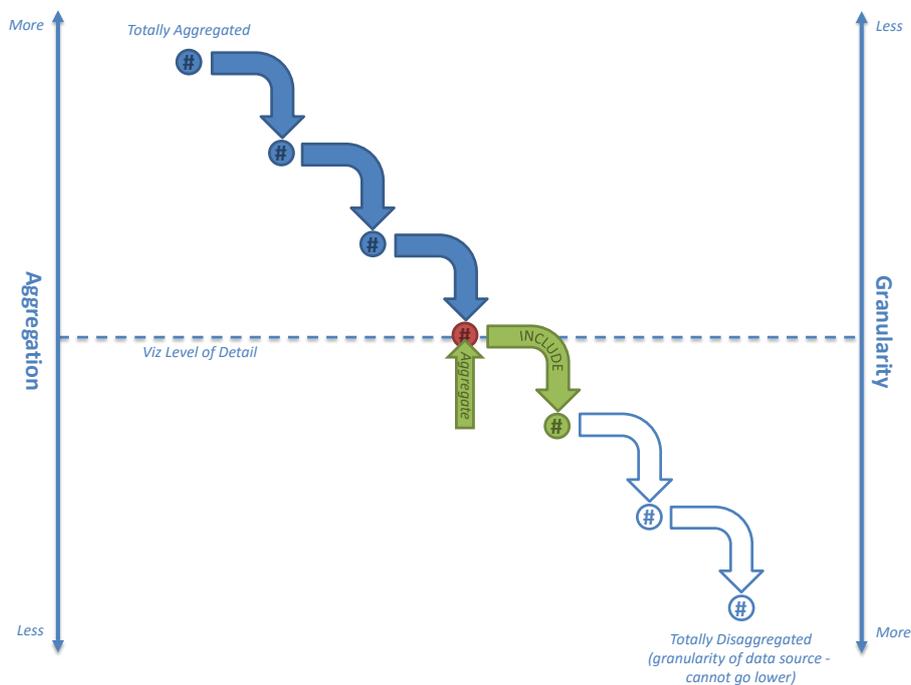
Test #	ATTR	MIN
1	2.824	1.857
2	2.62	2.09
3	2.737	2.878
4	2.804	1.977
5	2.994	1.882
Average	2.7958	2.1368
% improvement		24%

Weitere Informationen über ATTR() finden Sie im folgenden Blogbeitrag von InterWorks:

<http://bit.ly/1YEuZhX>

Detailgenaue Berechnungen

Mit LOD-Ausdrücken können Sie die Detailgenauigkeit definieren, mit der eine Berechnung ausgeführt werden soll, unabhängig von der Detailgenauigkeit der Visualisierung.



LOD-Ausdrücke bieten (in einigen Fällen) die Möglichkeit, Berechnungen zu ersetzen, die sie zuvor in früheren Versionen von Tableau auf mühsamere Weise verfasst haben:

- Mit Tabellenberechnungen haben Sie vielleicht versucht, die erste oder letzte Periode einer Partition zu finden. Wenn Sie zum Beispiel die Beschäftigtenzahl einer Organisation am ersten Tag eines jeden Monats berechnen.
- Mit Tabellenberechnungen, berechneten Feldern und Referenzlinien haben Sie vielleicht versucht, aggregierte Felder zu partitionieren. Zum Beispiel die Bestimmung des Mittelwerts einer bestimmten Kundenanzahl.
- Mithilfe der Datenverschmelzung haben Sie vielleicht versucht, eine relative Datenfilterung in Bezug auf das höchste Datum in den Daten zu erreichen. Wenn Ihre Daten zum Beispiel wöchentlich aktualisiert werden, die Berechnung der Summen des bisherigen laufenden Jahres in Bezug auf das höchste Datum.

Anders als Tabellenberechnungen werden LOD-Ausdrücke als Teil einer Abfrage zur zugrunde liegenden Datenquelle generiert. Sie werden als eine verschachtelte Auswahl ausgedrückt, sind also abhängig von der DBMS-Leistung:

```
SELECT T1.[State],SUM(T2.[Sales per County, State])
FROM [DB Table] T1 INNER JOIN
    (SELECT [State], [County], AVG([Sales]) AS [Sales per County, State]
    FROM [DB Table] GROUP BY [State],[County]) T2
ON T1.[State] = T2.[State]
GROUP BY T1.[State]
```

Dies bedeutet, dass es in einigen Szenarios möglicherweise effizienter ist, das Problem mit dem einen oder dem anderen Ansatz zu lösen: Z. B. kann eine Tabellenberechnung oder eine Verschmelzung eine bessere Leistung erbringen, als ein LOD-Ausdruck, oder umgekehrt. Wenn Sie vermuten, dass die Leistung aufgrund eines LOD-Ausdrucks vermindert ist, können Sie versuchen, sie durch eine Tabellenberechnungen oder eine Datenverschmelzung (sofern möglich) zu ersetzen, um festzustellen, ob sich die Leistung verbessert. LOD-Ausdrücke können auch stark von einer Verknüpfungsauswahl beeinträchtigt werden. Deshalb empfehlen wir, dass Sie den betreffenden Abschnitt noch einmal lesen, falls Sie den Eindruck haben, dass Ihre Abfragen langsam ablaufen, wenn LOD-Ausdrücke verwendet werden.

Um die Mechanik der LOD-Ausdrücke besser zu verstehen, lesen Sie das Whitepaper „Einführung in LOD-Ausdrücke“:

<http://www.tableau.com/de-de/learn/whitepapers/understanding-lod-expressions>

Sie können auch den Blogbeitrag „Top 15 LOD-Ausdrücke“ von Bethany Lyons lesen. Er enthält Beispiele für eine Reihe häufig auftretender Probleme:

<http://www.tableau.com/de-de/about/blog/LOD-expressions>

Schließlich gibt es eine Vielzahl von Blogbeiträgen aus der Community zu diesem Thema, die äußerst hilfreich sind. Eine hervorragende kuratierten Liste ist verfügbar unter *data + science*:

<http://bit.ly/1MpkFV5>

Tabellenberechnungen

Tabellenberechnungen werden anders als Berechnungen auf Zeilenebene und aggregierte Berechnungen nicht von der Datenbank ausgeführt, sondern im Abfrageergebnissatz von Tableau berechnet. Dies bedeutet zwar mehr Arbeit für Tableau, wird jedoch im Allgemeinen an einem im Vergleich zur ursprünglichen Datenquelle wesentlich kleineren Datensatz ausgeführt.

Wenn die Leistung der Tabellenberechnung ein Problem ist (möglicherweise, weil der an Tableau zurückgegebene Ergebnissatz sehr groß ist), überprüfen Sie die Möglichkeit, einige Aspekte der Berechnung auf die Datenquellenebene zurückzuverlagern. Eine Möglichkeit, dies zu tun, ist ein LOD-Ausdruck. Eine andere Möglichkeit wäre, einen aggregierten Datenextrakt zu nutzen. Stellen Sie sich ein Beispiel vor, bei dem Sie das Wochenmittel der Tagesumsätze mehrerer Geschäfte ermitteln möchten. Dies können Sie mit einer Tabellenberechnung mithilfe dieser Funktion erreichen:

```
WINDOW_AVG(SUM([Sales]))
```

Wenn die Anzahl der Tage bzw. Geschäfte jedoch sehr groß ist, kann diese Berechnung langsam werden. Um SUM([Umsatz]) auf die Datenebene zurückzuverlagern, erstellen Sie einen aggregierten Extrakt, der die Datumsdimension bis zur Tagesebene aufrechnet. Die Berechnung kann dann einfach mit AVG([Umsatz]) erfolgen, da der Extrakt bereits die Tagessummen enthält.

In einigen Anwendungsfällen wissen wir möglicherweise, dass sich der Wert des Aggregatsfaktors in der Partition nicht ändert. Wenn dies der Fall ist, verwenden Sie MIN() oder MAX() als Aggregatfunktion, anstelle von AVG() oder ATTR(), weil sie schneller evaluiert werden. Und wenn Sie eine ausgewählt haben, behalten Sie sie bei, damit die Cache-Trefferquote erhöht wird.

Externe Berechnungen

Tableau bietet Mechanismen, mit denen die komplexe Logik zur Berechnung an eine externe Engine übergeben werden kann. Etwa Aufrufe in R (über eine Rserve-Verbindung) oder Python (über eine Dato-Serververbindung).

Innerhalb von Tableau ähneln diese Vorgänge den Tabellenberechnungen, sie werden also partitionsübergreifend berechnet. Das bedeutet, dass sie für eine Visualisierung mehrfach aufgerufen werden können, was eine Leistungsbeeinträchtigung zur Folge haben kann. Sie werden auch weder optimiert noch verschmolzen. Überlegen Sie also, ob Sie mehrere Rückgabewerte mit einem einzigen Funktionsaufruf erhalten können (z. B. in einer verketteten Zeichenfolge), anstatt mehrere Funktionsaufrufe auszuführen. Schließlich kann auch die Zeit ausschlaggebend sein, die für die Datenübertragung zu/von der externen Berechnung-Engine erforderlich ist.

Analysen

Über den Bereich „Analysen“ können Sie schnell auf eine Reihe fortgeschrittener Analysefunktionen zugreifen, etwa:

- Gesamtwerte
- Referenzlinien
- Trendlinien
- Prognosen
- Clustering (neu in Tableau 10)

Für diese Analysen sind im Allgemeinen keine zusätzlichen Abfragen erforderlich. Sie werden eher wie Tabellenberechnungen mit den Daten im Abfrageergebnis-Cache ausgeführt. Wie bei Tabellenberechnungen gilt: Wenn die Abfrageergebnisse einen sehr großen Datenbestand enthalten, kann die Berechnung einige Zeit in Anspruch nehmen, aber im Allgemeinen trägt dies nicht wesentlich zu schlechter Arbeitsmappenleistung bei.

Ein weiterer Faktor bei der Leistung mit Gesamtwerten und Referenzlinien ist, ob die Kennzahlaggregationen additiv sind oder nicht. Bei additiven Aggregationen (z. B. SUM, MIN, MAX) kann die Berechnung von Gesamtwerten oder Referenzlinien lokal in Tableau vorgenommen werden.

Bei nicht additiven Aggregationen (z. B. COUNTD, TOTAL, MEDIAN, PERCENTILE) müssen wir auf die Datenquelle zurückgreifen, um die Gesamtwerte bzw. Referenzlinien zu berechnen.

Berechnungen im Vergleich zu nativen Funktionen

Gelegentlich erstellen Benutzer berechnete Felder zum Ausführen von Funktionen, obwohl diese mithilfe nativer Funktionen von Tableau einfach erzielt werden können. Beispiel:

- Gruppieren von Dimensionselementen – ziehen Sie [Gruppen](#) oder [Sätze](#) in Betracht;
- Gruppieren von Kennzahlwerten in Bereichsbändern – ziehen Sie [Partitionen](#) in Betracht;
- Verkürzen von Datumsangaben auf eine gröbere Granularität zum Beispiel Monat oder Woche – ziehen Sie [benutzerdefinierte Datumsfelder](#) in Betracht;
- Erstellen eines Wertesatzes durch Verkettung von zwei verschiedenen Dimensionen – ziehen Sie [kombinierte Felder](#) in Betracht;
- Anzeigen von Datumsangaben in einem bestimmten Format oder Konvertieren numerischer Werte in KPI-Indikatoren – ziehen Sie die Verwendung integrierter Formatierungsfunktionen Betracht;
- Ändern der angezeigten Werte für Dimensionselemente: ziehen Sie [Aliase](#) in Betracht.

Dies ist nicht immer möglich (Sie benötigen vielleicht Partitionen variabler Breite und das ist bei grundlegenden Partitionen nicht möglich), aber ziehen Sie den Einsatz nativer Optionen so häufig wie möglich in Betracht. Dies ist häufig effizienter als eine manuelle Berechnung. Da unsere Entwickler die Leistung von Tableau weiterhin verbessern, werden Sie von ihren Anstrengungen profitieren.

Einfluss von Datentypen

Wenn berechnete Felder erstellt werden, beachten Sie, dass der verwendete Datentyp einen wesentlichen Einfluss auf die Berechnungsgeschwindigkeit hat. Als allgemeine Richtlinie gilt Folgendes:

- Ganzzahlen und boolesche Werte sind viel schneller als Zeichenfolgen und Datumsangaben

Zeichenfolgen- und Datumsberechnungen sind sehr langsam. Häufig müssen für jede Berechnung 10 bis 100 grundlegende Anweisungen ausgeführt werden. Im Vergleich dazu sind numerische und boolesche Berechnungen sehr effizient.

Dies trifft nicht nur auf die Berechnungs-Engine von Tableau zu, sondern auch auf die meisten Datenbanken. Da Berechnungen auf Zeilenebene und aggregierte Berechnungen zur Datenbank verlagert werden, wenn sie in Form einer numerischen anstatt einer Zeichenfolgelogik ausgedrückt sind, werden sie viel schneller ausgeführt.

Leistungstechniken

Ziehen Sie die folgenden Techniken in Betracht, um sicherzustellen, dass Ihre Berechnungen so effizient wie möglich sind:

Verwenden Sie boolesche Operationen für grundlegende logische Berechnungen

Wenn Sie eine Berechnung haben, die ein binäres Ergebnis produziert (zum Beispiel ja/nein, bestanden/nicht bestanden, über/unter), sollte ein boolesches Ergebnis zurückgegeben werden und keine Zeichenfolge. Beispiel:

```
IF [Date] = TODAY() THEN "Today"  
ELSE "Not Today"  
END
```

Dies wäre sehr langsam, weil Zeichenfolgen verwendet werden. Schneller wird es durch die Rückgabe eines booleschen Ergebnisses:

```
[Date] = TODAY()
```

Verwenden Sie dann Aliase, um die Ergebnisse TRUE und FALSE in „Today“ (Heute) und „Not Today“ (Nicht heute) umzubenennen.

Zeichenfolge-Suchvorgänge

Stellen Sie sich vor, Sie möchten die Möglichkeit haben, alle Datensätze anzuzeigen, in denen der Produktname die gesuchte Zeichenfolge enthält. Sie können einen Parameter verwenden, um die gesuchte Zeichenfolge vom Benutzer zu erhalten, und dann das folgende berechnete Feld einrichten:

```
IF FIND([Product Name],[Product Lookup]) > 0
  THEN [Product Name]
  ELSE NULL
END
```

Diese Berechnung ist langsam, denn es ist ineffizient, auf diese Weise nach enthaltenen Zeichenfolgen zu suchen. Eine bessere Möglichkeit wäre, die spezifische Funktion CONTAINS zu verwenden, weil diese nach Weitergabe an die Datenbank in optimale SQL umgewandelt wird:

```
CONTAINS([Product Name],[Product Lookup])
```

In diesem Fall wäre die beste Lösung allerdings, gar kein berechnetes Feld zu verwenden, sondern eine Filterkarte mit Platzhalterzeichen.

Parameter für bedingte Berechnungen

Eine häufig auftretende Technik in Tableau besteht darin, für Endbenutzer einen Parameter bereitzustellen, damit er einen Wert auswählen kann, der die Art der Berechnungsausführung bestimmt. Beispiel: Angenommen, der Endbenutzer soll die Ebene der Datumsaggregation für eine Ansicht steuern, indem er mögliche Werte aus einer Auswahlliste wählt. Viele würden einen Zeichenfolgeparameter erstellen:

```
Value
Year
Quarter
Month
Week
Day
```

Danach würden Sie ihn in einer Berechnung der folgenden Art verwenden:

```
CASE [Parameters].[Date Part Picker]
  WHEN "Year" THEN DATEPART('year',[Order Date])
  WHEN "Quarter" THEN DATEPART('quarter',[Order Date])
  ..
END
```

Eine bessere Lösung wäre, die integrierte Funktion DATEPART() zu verwenden und die Berechnung wie folgt zu erstellen:

```
DATEPART(LOWER([Parameters].[Date Part Picker]), [Order Date])
```

In älteren Versionen von Tableau war der letztere Ansatz viel schneller. Jetzt gibt es jedoch keinen Leistungsunterschied mehr zwischen diesen beiden Lösungen, weil wir die Bedingungslogik der CASE-Anweisung auf der Grundlage des Parameterwerts reduzieren und nur den geeigneten DATEPART() an die Datenquelle weitergeben.

Vergessen Sie nicht, dass Sie auch die zukünftige Wartungsfreundlichkeit der Lösung in Betracht ziehen müssen. Unter diesem Aspekt wäre die letztere Berechnung wahrscheinlich eine bessere Option.

Datumsumwandlung

Benutzer verfügen oft über Datumsangaben, die nicht im nativen Datumsformat gespeichert sind. Sie können also eine Zeichenfolge oder ein numerischer Zeitstempel sein. Die Funktion DATEPARSE() vereinfacht diese Umwandlungen. Sie können einfach eine Formatierungszeichenfolge übergeben:

```
DATEPARSE("yyyymmdd", [YYYYMMDD])
```

Beachten Sie, dass DATEPARSE() nur von einer Teilmenge der Datenquellen unterstützt wird:

- nicht veraltete Excel- und Textdateiverbindungen
- MySQL
- Oracle
- PostgreSQL
- Tableau-Datenextraktion

Wenn Ihre Datenquelle DATEPARSE() nicht unterstützt, besteht die alternative Technik zum Konvertieren in ein ordnungsgemäßes Tableau-Datum darin, das Feld in eine Datumszeichenfolge zu parsen (zum Beispiel „2012-01-01“ – beachten Sie, dass ISO-Zeichenfolgen zu bevorzugen sind, weil sie der Internationalisierung entsprechen), und es anschließend an die Funktion DATE() weiterzugeben.

Wenn es sich bei den Ursprungsdaten um ein numerisches Feld handelt, ist es sehr ineffizient, in eine Zeichenfolge umzuwandeln und anschließend in ein Datum. Es ist viel besser, die Daten numerisch zu lassen und DATEADD() sowie literale Datumswerte zum Ausführen der Berechnung zu verwenden.

Eine langsame Berechnung wäre beispielsweise Folgendes (Konvertierung eines Zahlenfeldes im ISO-Format):

```
DATE(LEFT(STR([YYYYMMDD]), 4)
+ "-" + MID(STR([YYYYMMDD]), 4, 2)
+ "-" + RIGHT(STR([YYYYMMDD]), 2))
```

Für diese Berechnung wäre die folgende Vorgehensweise effizienter:

```
DATEADD('day', INT([yyyymmdd])% 100 - 1,
DATEADD('month', INT([yyyymmdd]) % 10000 / 100 - 1,
DATEADD('year', INT([yyyymmdd]) / 10000 - 1900,
#1900-01-01#))
```

Eine noch bessere Lösung wäre, die Funktion MAKEDATE() zu verwenden, wenn sie von Ihrer Datenquelle unterstützt wird:

```
MAKEDATE(2004, 4, 15)
```

Beachten Sie, dass der Leistungsgewinn bei großen Datensätzen erheblich sein kann. Bei einer Datenbank mit mehr als 1 Milliarde Datensätzen benötigte die erste Berechnung mehr als 4 Stunden, während die zweite 1 Minute dauerte.

Logische Anweisungen

Zunächst testen, welches Ergebnis am häufigsten auftritt

Wenn Tableau einen logischen Test evaluiert, stoppt es die Durcharbeitung der möglichen Ergebnisse, sobald eine Übereinstimmung gefunden wurde. Das bedeutet, dass Sie zunächst das

Ergebnis mit der größten Wahrscheinlichkeit ermitteln sollten, sodass die Evaluation oft schon nach dem ersten Test abgeschlossen ist.

Sehen Sie sich das folgende Beispiel an. Diese Logik:

```
IF <unlikely_test>
  THEN <unlikely_outcome>
  ELSEIF <likely_test>
    THEN <likely_outcome>
  END
```

Erfordert mehr Zeit zum Evaluieren als diese:

```
IF <likely_test>
  THEN <likely_outcome>
  ELSEIF <unlikely_test>
    THEN <unlikely_outcome>
  END
```

ELSEIF > ELSE IF

Wenn Sie mit komplexen logischen Anweisungen arbeiten, bedenken Sie Folgendes: ELSEIF > ELSE IF. Der Grund dafür ist, dass eine verschachtelte IF-Anweisung in der daraus resultierenden Abfrage als verschachtelte CASE-Anweisung und nicht als Teil der ersten berechnet wird. Somit würde dieses berechnete Feld:

```
IF [Region] = "East" AND [Segment] = "Consumer"
  THEN "East-Consumer"
  ELSE IF [Region] = "East" and [Segment] <> "Consumer"
    THEN "East-All Others"
  END
END
```

Den folgenden SQL-Code mit zwei verschachtelten **CASE**-Anweisungen und 4 bedingten Tests produzieren:

```
(CASE
  WHEN (([Global Superstore].[Region] = 'East')
  AND ([Global Superstore].[Segment] = 'Consumer'))
  THEN 'East-Consumer'
  ELSE (CASE
    WHEN (([Global Superstore].[Region] = 'East')
    AND ([Global Superstore].[Segment] <> 'Consumer'))
    THEN 'East-All Others'
    ELSE CAST(NULL AS NVARCHAR)
  END)
END)
```

Dies läuft schneller ab, wenn es wie unten mit der ELSEIF-Anweisung umgeschrieben wird, statt mit verschachtelten IF-Anweisungen. Dies wäre auch eine effizientere Nutzung des bedingten Tests:

```
IF [Region] = "East" AND [Segment] = "Consumer"
  THEN "East-Consumer"
  ELSEIF [Region] = "East"
    THEN "East-All Others"
  END
```

Dies resultiert in nur einer CASE- Anweisung im SQL-Code:

```
(CASE
  WHEN (([Global Superstore].[Region] = 'East')
  AND ([Global Superstore].[Segment] = 'Consumer'))
  THEN 'East-Consumer'
  WHEN ([Global Superstore].[Region] = 'East')
```

```

        THEN 'East-All Others'
    ELSE CAST(NULL AS NVARCHAR)
END)

```

Aber diese Form ist noch schneller. Sie verwendet zwar eine verschachtelte IF-Anweisung, nutzt jedoch die bedingten Tests am effizientesten:

```

IF [Region] = "East" THEN
    IF [Segment] = "Consumer"
        THEN "East-Consumer"
        ELSE "East-All Others"
    END
END

```

Der daraus resultierende SQL-Code:

```

(CASE
    WHEN ([Global Superstore].[Region] = 'East')
    THEN (CASE
        WHEN ([Global Superstore].[Segment] = 'Consumer')
        THEN 'East-Consumer'
        ELSE 'East-All Others'
    END)
    ELSE CAST(NULL AS NVARCHAR)
END)

```

Redundante Logikprüfungen vermeiden

Entsprechend wird empfohlen, redundante Logikprüfungen zu vermeiden. Die folgende Berechnung:

```

IF [Sales] < 10 THEN "Bad"
ELSEIF [Sales] >= 10 AND [Sales] < 30 THEN "OK"
ELSEIF [Sales] >= 30 THEN "Great"
END

```

Generiert den folgenden SQL-Code:

```

(CASE
    WHEN ([Global Superstore].[Sales] < 10)
    THEN 'Bad'
    WHEN (([Global Superstore].[Sales] >= 10)
    AND ([Global Superstore].[Sales] < 30))
    THEN 'OK'
    WHEN ([Global Superstore].[Sales] >= 30)
    THEN 'Great'
    ELSE CAST(NULL AS NVARCHAR)
END)

```

Dies wäre in dieser Form effizienter:

```

IF [Sales] < 10 THEN "Bad"
ELSEIF [Sales] >= 30 THEN "Great"
ELSE "OK"
END

```

So ergibt sich der folgende SQL-Code:

```

(CASE
    WHEN ([Global Superstore].[Sales] < 10)
    THEN 'Bad'
    WHEN ([Global Superstore].[Sales] >= 30)
    THEN 'Great'
    ELSE 'OK'
END)

```

Die kleinen Dinge zählen

Es gibt zahlreiche kleine Maßnahmen, die Sie ergreifen können, und die sich auf die Leistung auswirken. Ziehen Sie die folgenden Tipps in Betracht:

- Der Umgang mit Datumslogik kann kompliziert sein. Anstatt ausgefeilte Tests mit mehreren Datumsfunktionen zu schreiben – z. B. MONTH() und YEAR() –, ziehen Sie die Verwendung einiger anderer integrierter Funktionen wie DATETRUNC(), DATEADD() und DATEDIFF() in Betracht. Dies kann die Komplexität der für die Datenquelle generierten Abfrage signifikant reduzieren.
- Verschiedene Zielwerte gehören zu den langsamsten Aggregationstypen bei fast allen Datenquellen. Verwenden Sie die Aggregation COUNTD nur sparsam.
- Wenn Sie Parameter mit großer Auswirkung verwenden (z. B. in einer benutzerdefinierten SQL-Anweisung), kann dies die Cache-Leistung beeinträchtigen.
- Das Filtern mit komplexen Berechnungen kann zur Folge haben, dass Indizes in der zugrunde liegenden Datenquelle nicht berücksichtigt werden.
- Verwenden Sie NOW() nur, wenn Sie die Detaildaten der Zeitstempelsebene benötigen. Verwenden Sie TODAY() für Berechnungen auf Datumsebene.
- Vergessen Sie nicht, dass alle grundlegenden Berechnungen über die zugrunde liegende Datenquelle laufen – sogar Literalberechnungen wie Beschriftungszeichenfolgen. Wenn Sie Beschriftungen erstellen müssen (z. B. für Spaltenüberschriften) und Ihre Datenquelle sehr groß ist, erstellen Sie eine einfache Text-/Excel-Datei-Datenquelle mit nur einem Datensatz dafür. Auf diese Weise verursachen sie keinen zusätzlichen Aufwand bei einer großen Datenquelle. Dies ist besonders wichtig, wenn Ihre Datenquelle gespeicherte Prozeduren verwendet – weitere Hinweise finden Sie in [diesem Abschnitt](#).

Liegt es an meinen Abfragen?

Eine der sehr leistungsstarken Funktionen von Tableau ist, dass es sowohl speicherresidente Daten (z. B. extrahierte Datenverbindungen) sowie Daten am Standort verwenden kann (etwa direkte Datenverbindungen). Direktverbindungen sind eine sehr leistungsstarke Funktion von Tableau, weil Sie damit die in der Datenquelle bereits vorhandene Verarbeitungsleistung nutzen können. Da wir jetzt jedoch hinsichtlich der Leistung von der Quelldatenplattform abhängig sind, ist es von entscheidender Bedeutung, unsere Abfragen in der Datenquelle so effizient und optimal wie möglich zu generieren.

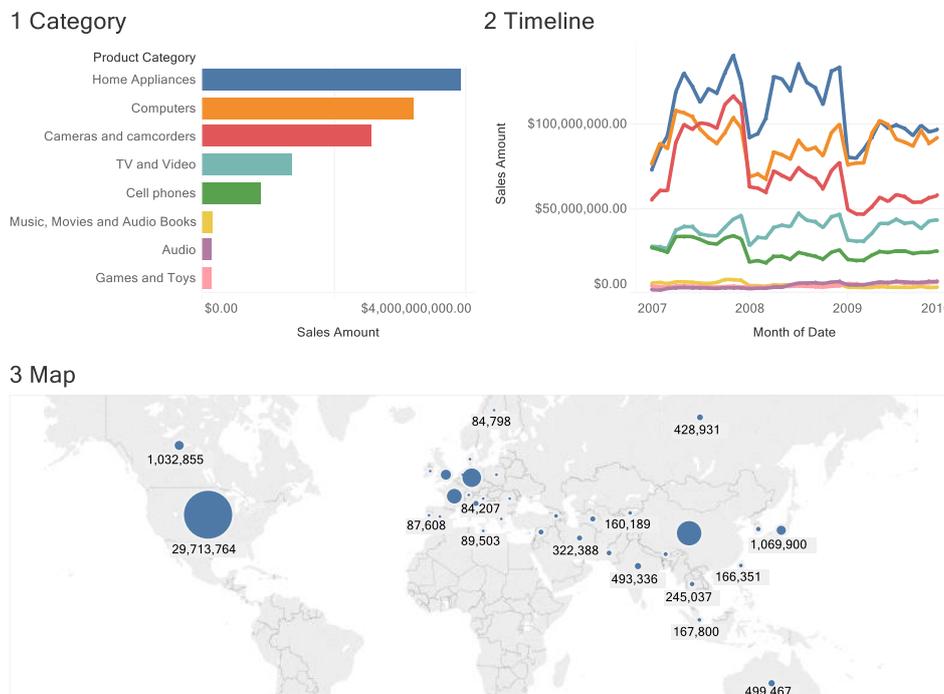
Wir haben es bereits erwähnt: Je komplexer die Frage, je mehr Daten Sie anzeigen, je mehr Elemente Sie in das Dashboard einbeziehen, desto mehr muss die Datenquelle arbeiten. Damit unsere Arbeitsmappen so effizient wie möglich werden, müssen wir die Anzahl der Abfragen minimieren, sicherstellen, dass die Abfragen so effizient wie möglich evaluiert werden, den Umfang der von den Abfragen zurückgegebenen Daten minimieren und zwischen den Anfragen so viele Daten wie möglich wiederverwenden.

Automatische Optimierungen

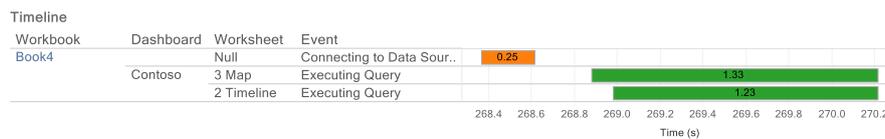
Wir können zwar viele Dinge tun, um die Leistung unserer Abfragen zu verbessern, allerdings gibt es viele Optimierungen, die Tableau automatisch durchführt, damit Ihre Arbeitsmappen effizient ausgeführt werden. Die meisten dieser Maßnahmen können Sie nicht direkt steuern und in den meisten Fällen brauchen Sie nicht darüber nachzudenken. Wenn Sie sie jedoch umfassend verstehen, können Sie sie zur Leistungssteigerung nutzen.

Parallelausführung – mehrere Abfragen gleichzeitig ausführen

Tableau nutzt die Fähigkeit der Quelldatenbanken, mehrere Abfragen gleichzeitig auszuführen. Betrachten Sie das folgende Dashboard:



Wenn Sie diese Arbeitsmappe in Tableau öffnen, sehen Sie die folgenden Abfragen:



Wie Sie sehen, würde die serielle Ausführung der Abfragen 2,66 Sekunden in Anspruch nehmen. Wenn Sie jedoch parallel ausgeführt werden, benötigen Sie nur die Zeit der längsten Abfrage (1,33 Sekunden).

Das Niveau der Abfragenparallelität wechselt je nach Quellsystem, weil einige Plattformen gleichzeitige Abfragen besser verarbeiten als andere. Standardmäßig werden bei allen Datenquellen außer Text, Excel und Statistikdateien maximal 16 Parallelabfragen ausgeführt (Grenzwert ist eine gleichzeitige Abfrage). Bei anderen Daten wurde der Grenzwert möglicherweise unter dem Standardwert festgelegt. Details finden Sie über den folgenden Link:

<http://kb.tableau.com/articles/HowTo/Configuring-Parallel-Queries-in-Tableau-Desktop?lang=de-de>

In den meisten Fällen brauchen diese Einstellungen nicht geändert zu werden. Behalten Sie die Standardeinstellungen bei. Wenn Sie den Parallelitätsgrad aus einem bestimmten Grund ändern müssen, haben Sie folgende Möglichkeiten:

- Eine globale Begrenzung der Anzahl der Parallelabfragen für Tableau Server
- Grenzen für einen bestimmten Datenquellentyp, etwa SQL-Server
- Grenzen für einen bestimmten Datenquellentyp auf einem bestimmten Server
- Grenzen für einen bestimmten Datenquellentyp auf einem bestimmten Server, wenn eine Verbindung zu einer bestimmten Datenbank hergestellt wird

Diese Einstellungen werden über eine xml-Datei mit dem Namen connection-configs.xml verwaltet, die Sie erstellen und im Ordner „App“ unter Windows (C:\Program Files\Tableau\Tableau 10.0) und auf dem Mac (mit der rechten Maustaste auf die Anwendung klicken, dann „Paketinhalt anzeigen“ und die Datei dort ablegen) für Tableau Desktop oder im Verzeichnis „config“ im Ordner „vizqlserver“ (zum Beispiel:

C:\ProgramData\Tableau\TableauServer\data\tabsvc\config\vizqlserver) für Tableau Server speichern können. Sie müssen diese Konfigurationsdatei in alle vizqlserver-Konfigurationsverzeichnisse auf allen Arbeitsservern kopieren.

Weitere Hinweise zum Konfigurieren von Parallelabfragen, einschließlich der Syntax der Datei connection-configs.xml finden Sie hier:

<http://kb.tableau.com/articles/knowledgebase/parallel-queries-tableau-server?lang=de-de>

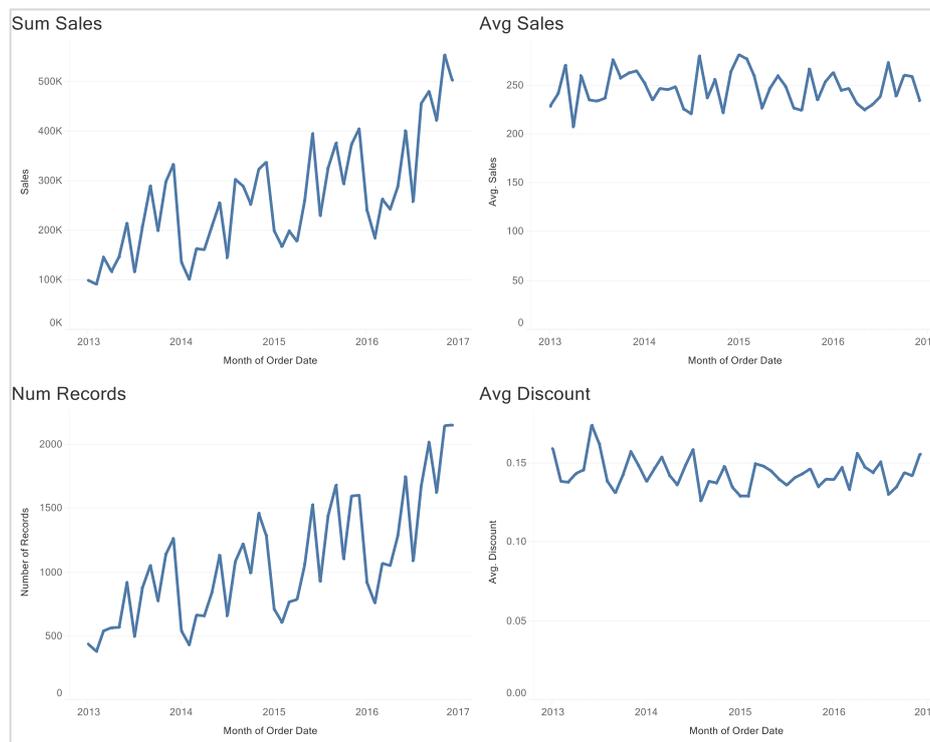
Abfragen eliminieren – weniger Abfragen ausführen

Im Beispiel oben sehen Sie auch, dass wir nur zwei Abfragen anstatt drei ausgeführt haben.

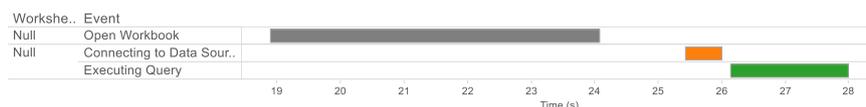
Durch Zusammenführung von Abfragen kann Tableau redundante Abfragen eliminieren.

Der Abfrageoptimierer von Tableau sortiert die Abfragen so, dass die komplexesten zuerst ausgeführt werden, und zwar in der Hoffnung, dass nachfolgende Abfragen aus dem Ergebnis-Cache bedient werden können. Im Beispiel können die Daten für das Kategoriediagramm aus dem Abfrage-Cache des Arbeitsblatts „Zeitachse“ abgerufen werden, weil die Zeitachse „Produktkategorie“ enthält und weil die SUM-Aggregation von „Umsatzbetrag“ vollständig additiv ist. Es ist also nicht erforderlich auf die Datenquelle zuzugreifen.

Der Abfrageoptimierer sucht also nach Abfragen mit gleicher Detailgenauigkeit (z. B. mit dem gleichen Satz Dimensionen) und reduziert sie auf eine einzelne Abfrage, die alle angeforderten Kennzahlen zurückgibt. Betrachten Sie das folgende Dashboard:



Wie Sie sehen, enthält dieses Dashboard vier Blätter: Auf jedem wird eine andere Kennzahl im Zeitverlauf gezeigt. Sie haben alle die gleiche Detailgenauigkeit, weil sie die Daten anhand eines kontinuierlichen Monats zeigen. Anstatt vier getrennte Anfragen an die Datenbank zu senden, kombiniert Tableau sie zu einer einzigen Abfrage:



Die Abfrage lautet wie folgt:

```
SELECT AVG(cast([Global Superstore].[Discount] as float)) AS
[avg:Discount:ok],
AVG(cast([Global Superstore].[Sales] as float)) AS [avg:Sales:ok],
SUM(CAST(1 as BIGINT)) AS [sum:Number of Records:ok],
SUM([Global Superstore].[Sales]) AS [sum:Sales:ok],
DATEADD(month, DATEDIFF(month, CAST('0001-01-01 00:00:00' AS datetime2),
[Global Superstore].[Order Date]), CAST('0001-01-01 00:00:00' AS datetime2))
AS [tmn:Order Date:ok]
FROM [dbo].[Global Superstore] [Global Superstore]
GROUP BY DATEADD(month, DATEDIFF(month, CAST('0001-01-01 00:00:00' AS
datetime2), [Global Superstore].[Order Date]), CAST('0001-01-01 00:00:00' AS
datetime2))
```

Wie Sie sehen, kann diese Optimierung (auch „Abfrageverschmelzung“ genannt) die allgemeine Leistung signifikant verbessern. Sofern möglich, denken Sie immer daran, für mehrere Blätter zu einem Dashboard die gleiche Detailgenauigkeit zu definieren.

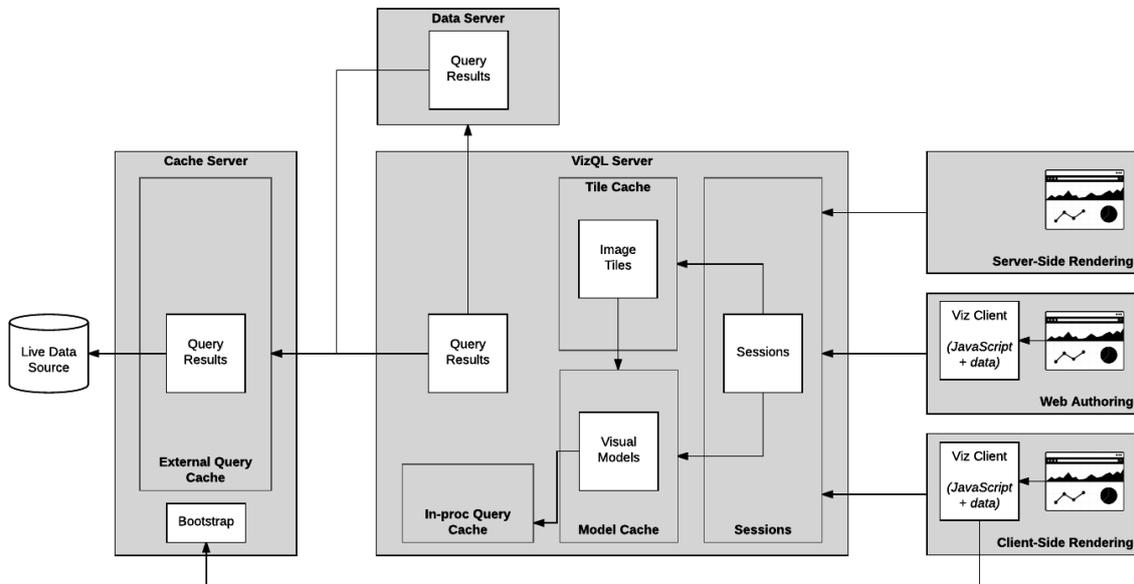
Beachten Sie, dass die Abfrageverschmelzung bei Abfragen für TDE-Datenquellen nicht ausgeführt wird.

Caching – gar keine Abfragen ausführen

Was ist besser als wenige Abfragen? Gar keine Abfragen! Tableau setzt den Cache-Speicher sowohl in Tableau Desktop als auch in Tableau Server umfassend ein. So kann die Anzahl der Abfragen in der zugrunde liegenden Datenquelle deutlich reduziert werden.

Tableau Server

Die Zwischenspeicherung von Tableau Server erfolgt auf mehreren Ebenen:



Die erste Cache-Ebene ist davon abhängig, ob die Sitzung clientseitiges oder serverseitiges Rendern einsetzt. In diesem [früheren Abschnitt](#) finden Sie weitere Informationen über die zwei Modelle zum Rendern und wie Tableau zwischen ihnen entscheidet.

Wenn die Sitzung clientseitiges Rendern verwendet, muss der Browser zunächst den Viewer-Client laden. Dabei werden ein Satz JavaScript-Bibliotheken und die Daten geladen, damit eine erste Ansicht gerendert werden kann. Dieser Vorgang wird „Bootstrapping“ genannt und kann einige Sekunden Anspruch nehmen. Da Dashboards wahrscheinlich von mehreren Benutzern angezeigt werden und um die Wartezeit für anschließende Anzeigeanforderungen zu reduzieren, werden die Bootstrap-Antworten zwischengespeichert. Jede Sitzung prüft zunächst den Bootstrap-Cache, um zu sehen, ob es schon eine Antwort gibt, die wiederverwendet werden kann. Wenn eine gefunden wird, lädt der Browser die Antwort einfach aus dem Cache, sodass die Ausgangsansicht sehr schnell gerendert werden kann. Sobald der Viewer-Client in den Client-Browser geladen ist, können einige Interaktionen nur mit den lokalen Daten ausgeführt werden (z. B. Hervorheben, QuickInfo), was in einer schnellen, reibungslosen Nutzererfahrung resultiert.

Wenn das Dashboard über serverseitiges Rendern angezeigt wird, rendert der Server die Dashboard-Elemente als eine Serie von Bilddateien („Kacheln“ genannt). Diese Kacheln werden an den Browser weitergeleitet, wo sie für die Anzeige der Visualisierung zusammengesetzt werden. Wie gesagt, ist zu erwarten, dass Dashboards von mehreren Benutzern aufgerufen werden. Der Server speichert die Bilder also im Cache auf der Festplatte. Bei jeder Anfrage wird der Kachel-Zwischenspeicher geprüft, um zu sehen, ob das Bild bereits gerendert wurde. Falls eins gefunden wird, wird einfach die Datei aus dem Cache geladen. Ein Treffer im Kachel-Zwischenspeicher beschleunigt die Reaktionszeit

und reduziert die Verarbeitungslast auf dem Server. Eine einfache Möglichkeit, um die Effizienz des Kachel-Zwischenspeichers zu erhöhen, besteht darin, die Dashboards mit [fester Größe](#) zu konzipieren.

Insgesamt sorgt das clientseitige Rendern für reibungslosere, reaktionsschnellere Benutzerinteraktionen und belastet den Tableau Server weniger. Sofern möglich, sollten Sie Ihre Arbeitsmappen so konzipieren, dass sie clientseitig gerendert werden können.

Die nächste Cache-Ebene wird „visuelles Modell“ genannt. Ein visuelles Modell beschreibt, wie ein einzelnes Arbeitsblatt gerendert werden soll. Die Anzeige eines Dashboards kann also mehrere visuelle Modelle verwenden – eines für jedes verwendete Arbeitsblatt. Das umfasst die Ergebnisse von lokalen Berechnungen (z. B. Tabellenberechnungen, Referenzlinien, Prognosen, Clustering usw.) und das visuelle Layout (wie viele Zeilen/Spalten für Rasterdiagramme und Kreuztabellen angezeigt werden sollen, Anzahl und Intervall der einzuzeichnenden Achsenteilstriche/Rasterlinien, Anzahl und Ort der angezeigten Markierungsbeschriftungen usw.).

Visuelle Modelle werden vom VizQL Server erstellt und im Zwischenspeicher abgelegt. Soweit möglich wird versucht, die Ergebnisse für verschiedene Benutzersitzungen zu verwenden. Die Schlüsselfaktoren dafür, ob ein visuelles Modell wiederverwendet werden kann, sind:

- Die Größe des Anzeigebereichs der Visualisierung: Modelle können nur für Sitzungen mit Visualisierungen identischer Größe wiederverwendet werden. Wenn Sie Dashboards mit fester Größe einrichten, profitieren Sie auf ähnliche Weise vom Modell-Zwischenspeicher wie vom Kachel-Zwischenspeicher: bessere Wiederverwendbarkeit und weniger Arbeitslast auf dem Server.
- Ob die Auswahlen/Filter übereinstimmen: Wenn der Benutzer Filter, Parameter ändert, Detaildaten aufruft usw., wird das Modell nur für solche Sitzungen wiederverwendet, deren Ansichtstatus übereinstimmt. Versuchen Sie, keine Arbeitsmappen mit aktivierter Option „Auswahl anzeigen“ zu veröffentlichen, weil damit die Wahrscheinlichkeit reduziert wird, dass verschiedene Sitzungen übereinstimmen.
- Die Anmeldedaten, die verwendet werden, um eine Verbindung mit der Datenquelle herzustellen: Wenn Benutzer aufgefordert werden, Anmeldedaten für die Verbindung zur Datenquelle einzugeben, kann das Modell nur für Benutzersitzungen mit übereinstimmenden Anmeldedaten wiederverwendet werden. Verwenden Sie diese Funktion mit Bedacht, weil sie die Effizienz des Modell-Zwischenspeichers reduzieren kann.
- Ob Benutzerfilterung verwendet wird: Wenn die Arbeitsmappe Benutzerfilter enthält oder Berechnungen mit Funktionen wie USERNAME() oder ISMEMBEROF(), wird das Modell nicht für andere Benutzersitzungen wiederverwendet. Verwenden Sie diese Funktionen mit Bedacht, denn sie können die Effizienz des Modell-Zwischenspeichers deutlich reduzieren.

Die letzte Cache-Ebene ist der Abfrage-Zwischenspeicher. Hier werden Ergebnisse von Abfragen gespeichert, die in Erwartung dessen ausgeführt wurden, dass sie zukünftige Abfragen bedienen können. Treffer in diesem Zwischenspeicher sind sehr effektiv, weil damit vermieden wird, dass Abfragen in der Datenquelle mehrfach ausgeführt werden. Die Daten werden einfach aus dem Zwischenspeicher geladen. In einigen Fällen können wir auch Abfragen mithilfe der Ergebnisse einer anderen Abfrage bedienen. Wir haben diese Vorteile [weiter oben](#) diskutiert, als wir die Abfrageneliminierung und die Abfragenverschmelzung erörtert haben.

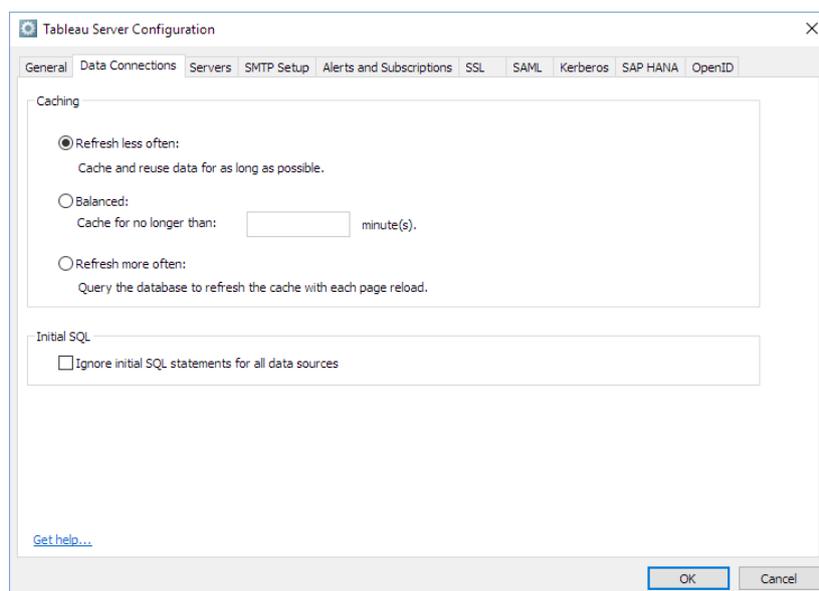
Dieser Zwischenspeicher hat zwei Teile: Einer befindet sich im VizQL-Prozess („In-Proc Cache“) und der andere wird über den Cache-Server für mehrere Prozesse freigegeben („Externer Cache“). Wenn eine Frage an eine VizQL-Instanz gesendet wird, die dieselbe Abfrage zuvor ausgeführt hat,

wird die Anfrage aus dem In-Proc Cache bedient. Beachten Sie, dass dieser Zwischenspeicher für jeden VizQL-Prozess lokal und speicherresident ist.

Zusätzlich zu diesen In-Proc Caches gibt es einen externen Cache, auf den nicht nur die VizQL-Instanzen, sondern ALLE Prozesse, die Zugriff auf die zugrunde liegende Datenquelle haben (z. B. Hintergrundprogramme, Datenserver usw.) zugreifen können. Ein Service mit dem Namen Cache-Server verwaltet den externen Cache für den gesamten Cluster. Beachten Sie, dass nicht alle Abfragen so geschrieben sind, dass sie den externen Cache nutzen. Wenn eine Abfrage sehr schnell abläuft, kann es weniger Zeit erfordern, sie erneut auszuführen, anstatt den Zwischenspeicher zu prüfen. Es gibt also eine Untergrenze für die Abfragezeit. Wenn das Ergebnis einer Abfrage sehr groß ist, kann es ineffizient sein, sie in den Cache-Server zu schreiben. Es gibt also auch eine Obergrenze.

Der externe Cache erhöht die Effektivität des Zwischenspeichers signifikant bei Bereitstellungen mit mehreren VizQL-Instanzen. Im Gegensatz zu In-Proc Cache, der flüchtig ist, ist der externe Cache dauerhaft (der Cache-Server schreibt die Daten auf die Festplatte) und bleibt zwischen Service-Instanzen erhalten.

Um die Effektivität des Zwischenspeichers zu maximieren, können Sie die Einstellungen der Tableau Server-Installation so anpassen, dass die Daten so lange wie möglich erhalten bleiben:



Wenn Sie verfügbare Server-Kapazitäten haben, können Sie die Cache-Größe erhöhen. Eine Erhöhung dieser Einstellungen hat keine Nachteile. Wenn Sie also ausreichend RAM haben, können Sie diese Werte deutlich erhöhen. So vermeiden Sie, dass Inhalte aus dem Cache gedrängt werden. Die Effektivität des Caches lässt sich über [Protokolldateien](#) oder über [TabMon](#) verfolgen.

- **Modell-Cache** – Standardmäßig werden 60 Modelle pro VizQL-Instanz gespeichert. Sie können diese Einstellung über den Befehl `tabadmin` anpassen. Wenn Sie RAM verfügbar haben, können Sie diesen Wert auf die Anzahl der auf Ihrem Server veröffentlichten Ansichten erhöhen:
 - `tabadmin set vizqlserver.modelcachesize 200`
- **In-Proc Cache** – Standardmäßig werden 512 MB Abfrageergebnisse pro VizQL-Instanz gespeichert. Dies klingt vielleicht nach wenig, aber bedenken Sie, dass Abfrageergebnisse aus aggregierten Abfragen zwischengespeichert werden. Sie können diese Einstellung über den Befehl `tabadmin` anpassen:
 - `tabadmin set vizqlserver.querycachesize 1024`

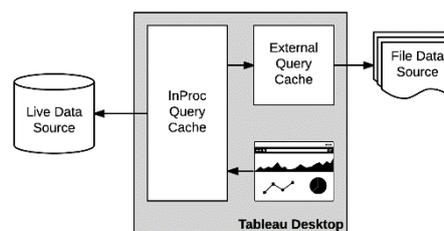
Sie können auch Kapazität und Durchsatz des externen Abfrage-Zwischenspeichers erhöhen, indem Sie weitere Instanzen des Cache-Servers hinzufügen. Es wird empfohlen, eine Instanz des Cache-Servers pro VizQL Server-Instanz auszuführen.

- **Externer Cache** – Standardmäßig werden 512 MB Abfrageergebnisse pro Cache-Server-Instanz gespeichert.

Ein nützlicher Nebeneffekt des geteilten Zwischenspeichers ist, dass wir den Cache für Arbeitsmappen vorwärmen können, von denen wir wissen, dass sie anfangs langsam sind, indem wir ein Abonnement für die Arbeitsmappe ausführen. Wenn Sie dies morgens einplanen, bevor Ihre interaktiven Benutzer am Arbeitsplatz sind, ist gewährleistet, dass die Arbeitsmappe ausgeführt wird und die Abfrageergebnisse in den Zwischenspeicher geladen werden. Angenommen, die Ergebnisse wurden nicht aus dem Cache gedrängt und sind nicht inzwischen abgelaufen, dann gibt es einen Treffer im Cache, wenn die Benutzer die Arbeitsmappe aufrufen, sodass die Ausgangsansicht schnell präsentiert wird.

Tableau Desktop

Die Cache-Vorgänge bei Tableau Desktop sind sehr viel einfacher als bei Tableau Server, weil es sich um eine Einzelbenutzeranwendung handelt, bei der es keine Mehrfachsitzungen gibt. In Tableau Desktop ist nur der Abfrage-Zwischenspeicher vorhanden:



Wie bei Tableau Server gibt es einen In-Proc- und einen externen Cache. Der In-Proc-Cache ist flüchtig, speicherresident und wird für Verbindungen zu allen Datenquellen verwendet. Der externe Cache wird jedoch nur für dateibasierte Datenquellen genutzt (z. B. Datenextrakte, Excel, Access, Textdateien, statistische Dateien usw.). Wie in Tableau Server ist der externe Cache dauerhaft. Das bedeutet, dass die Cache-Ergebnisse zwischen Tableau Desktop-Sitzungen beibehalten werden. Wenn Sie also eine Dateidatenquelle haben, können Sie Tableau Desktop neu starten und weiterhin vom Cache profitieren.

Der externe Cache-Speicher wird `%LOCALAPPDATA%\Tableau\Caching` unter Windows in und `~/Library/Caches/com.tableau/` auf dem Mac in gespeichert. Standardmäßig ist er auf insgesamt etwa 750 MB begrenzt und wird ungültig, wenn der Benutzer eine Aktualisierung der Datenquelle erzwingt (z. B. durch Drücken von F5, ⌘R).

Schließlich werden die Daten aus dem externen Abfrage-Zwischenspeicher einbezogen, wenn die Datei als Arbeitsmappenpaket gespeichert wird. So können Tableau Desktop und Tableau Reader die Ausgangsansicht der Arbeitsmappe schnell rendern, während die größere Datenquellendatei im Hintergrund dekomprimiert wird. Für den Endbenutzer verbessert dies die Reaktion beim Öffnen der Arbeitsmappe auf drastische Weise. Beachten Sie, dass die Cache-Daten nur einbezogen werden, wenn sie $\leq 10\%$ der Größe der Datenquellendatei ausmachen, und keine Abfragen vorhanden sind, die relative Datumsfilter verwenden.

Faule Verbindungen

Wenn ein Benutzer bei Versionen vor Tableau 10 eine Arbeitsmappe mit mehreren Datenquellen öffnete, wurde zunächst eine Verbindung zu allen Datenquellen ohne Anmeldedaten hergestellt (d. h. Datenquellen, für die der Benutzer weder Benutzernamen noch Kennwort eingeben muss). Das bedeutet, dass möglicherweise Zeit dafür aufgewendet wird, Verbindungen zu Datenquellen herzustellen, die auf dem Blatt/Dashboard/in der Story, die der Benutzer anzeigt, nicht verwendet werden.

In Tableau 10 stellen wir jetzt nur dann eine Verbindung zu einer Datenquelle her, wenn ein Benutzer sie für die Anzeige eines ausgewählten Blatts/Dashboards/einer ausgewählten Story benötigt. Diese Änderung reduziert die Ladezeit einer Arbeitsmappe mit Reitern (bei Tableau Desktop), sodass die Benutzer ihre Daten schneller untersuchen können.

Verknüpfungen

Der bevorzugte Ansatz für die Arbeit mit mehreren Tabellen in Tableau ist im Allgemeinen, die Verknüpfungen im Bereich „Datenverbindung“ zu definieren. Auf diese Weise definieren Sie nicht eine bestimmte Abfrage, sie definieren lediglich, welche Beziehungen zwischen den Tabellen bestehen. Wenn Sie Felder in eine Visualisierung ziehen, verwendet Tableau diese Informationen, um die dafür erforderliche Abfrage zu generieren, damit nur die benötigten Daten abgerufen werden.

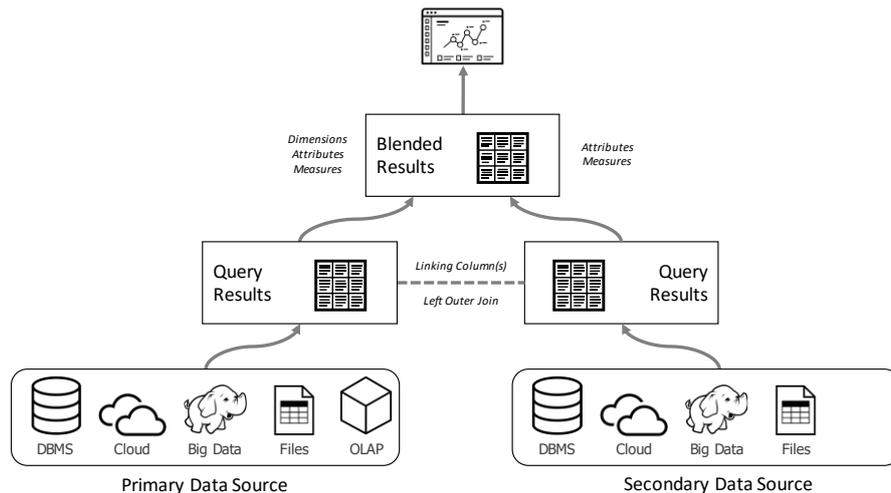
Als Faustregel gilt im Allgemeinen, dass die Anzahl der verknüpften Tabellen minimiert werden sollte, damit nur die Tabellen einbezogen werden, die für eine bestimmte Arbeitsmappe/ Visualisierung benötigt werden. Je nach Komplexität der Datenverbindung und Arbeitsmappen kann es sich also lohnen, Datenverbindungen für jede Arbeitsmappe sowie spezifische Verknüpfungsmuster für diese Blätter zu definieren.

Verknüpfungen funktionieren besser, wenn sie ordnungsgemäß eingeschränkt sind. Die Einschränkungen geben Tableau Desktop die Freiheit, Abfragen auf die Tabellen zu reduzieren, die zur Beantwortung bestimmter Teilmengen von Fragen erforderlich sind (Legenden, Filterkarten).

Verschmelzung

Wenn Sie eine Entscheidung treffen, ob Sie in Tableau Datentabellen verknüpfen oder Datentabellen verschmelzen, sollten Sie bedenken, wo die Daten herkommen, wie viele Datenverbindungen es gibt und wie hoch die Anzahl der Datensätze in den Daten ist.

Wenn Sie die Verschmelzung verwenden, ist einer der wichtigsten Einflussfaktoren auf die Leistung der Verschmelzung nicht die Anzahl von Datensätzen in jeder Datenquelle, sondern die Kardinalität der Verschmelzungsfelder, die die beiden Datensätze verbinden. Bei der Verschmelzung werden die Daten von beiden Datenquellen auf der Ebene der verknüpften Felder abgefragt und die Ergebnisse der beiden Abfragen werden im Arbeitsspeicher von Tableau zusammengefügt.



Wenn viele Einzelwerte vorhanden sind, kann dies sehr viel Arbeitsspeicher in Anspruch nehmen. Es wird empfohlen, für die Verschmelzung von Daten die 64-Bit-Version von Tableau Desktop zu verwenden, damit ausreichend Arbeitsspeicher zur Verfügung steht. Verschmelzungen dieser Art sind jedoch trotzdem zeitaufwendig.

Die Empfehlung aufgrund bewährter Verfahrensweisen bei der Verschmelzung besteht darin, ein Verschmelzen von Dimensionen mit einer großen Anzahl von Einzelwerten (z. B. Bestellnummer, Kundennummer, präzise Datums-/Zeitangaben usw.) zu vermeiden.

Primärgruppen und Aliase

Wenn Sie es für notwendig erachten, zwischen zwei Datenquellen eine Verschmelzung durchzuführen, weil eine die Faktendatensätze enthält und die andere Dimensionsattribute, kann es möglich sein, die Leistung durch die Schaffung einer Primärgruppe oder eines Alias zu verbessern.

Sie bilden Gruppen und/oder Aliase in der primären Datenquelle, um die in der sekundären Datenquelle aufgelisteten Attribute zu reflektieren. Primäre Gruppen werden für 1:Viele-Beziehungen verwendet und primäre Aliase für 1:1-Beziehungen. Sie können dann das relevante Objekt von der primären Datenquelle verwenden. Auf diese Weise ist es nicht erforderlich, beim Anzeigen zu verschmelzen.

Für die folgenden Beispiele betrachten Sie die folgenden drei Tabellen:

ID	Value
A	89
B	94
C	74
D	88
E	58
F	89
G	95

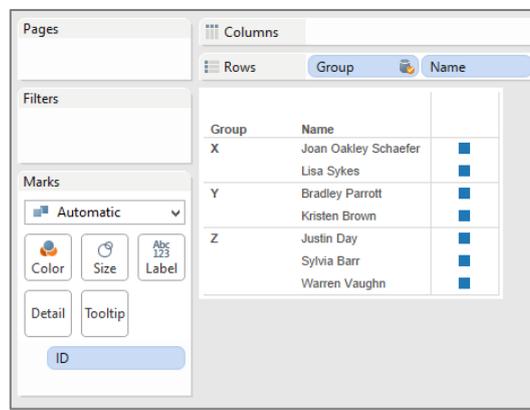
ID	Name
A	Lisa Sykes
B	Joan Oakley Schaefer
C	Bradley Parrott
D	Kristen Brown
E	Sylvia Barr
F	Warren Vaughn
G	Justin Day

ID	Group
A	X
B	X
C	Y
D	Y
E	Z
F	Z
G	Z

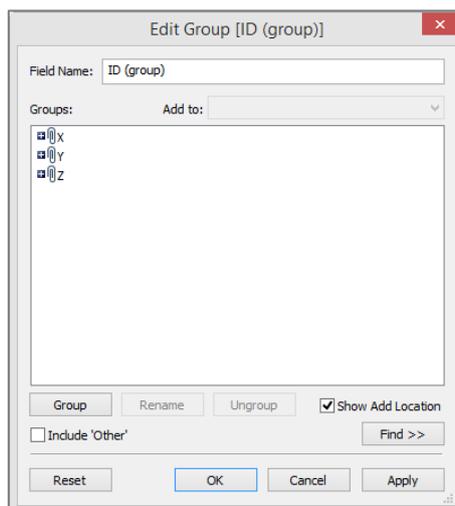
Primärgruppen sind nützlich, wenn die sekundäre Datenquelle ein Attribut enthält, das den Dimensionselementen in der primären Datenquelle im Verhältnis 1:n zugeordnet ist. Angenommen, aus den oben genannten Daten soll Folgendes erstellt werden:

Group	Name
X	Lisa Sykes
	Joan Oakley Schaefer
Y	Bradley Parrott
	Kristen Brown
Z	Sylvia Barr
	Warren Vaughn
	Justin Day

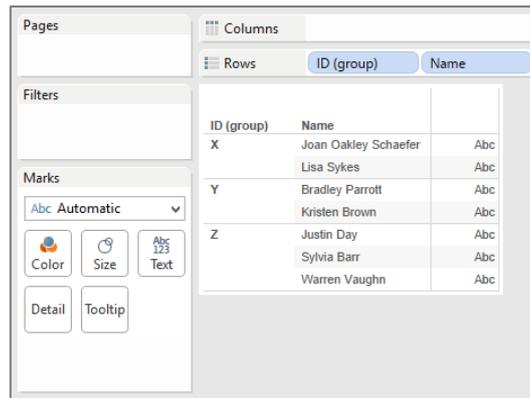
Dies kann durch Verschmelzung erfolgen. Aber wie bereits besprochen, würde dies zu einer sehr schwachen Leistung führen, wenn eine größere Anzahl von IDs vorhanden ist:



Durch einen Klick mit der rechten Maustaste auf das Feld „Gruppe“ und Auswahl von „Primärgruppe erstellen“ erstellt Tableau ein Gruppenobjekt in der primären Datenquelle, das das Verknüpfungsfeld (in diesem Fall „ID“) der ausgewählten Dimension der sekundären Datenquelle (in diesem Fall „Gruppe“) zuordnet.



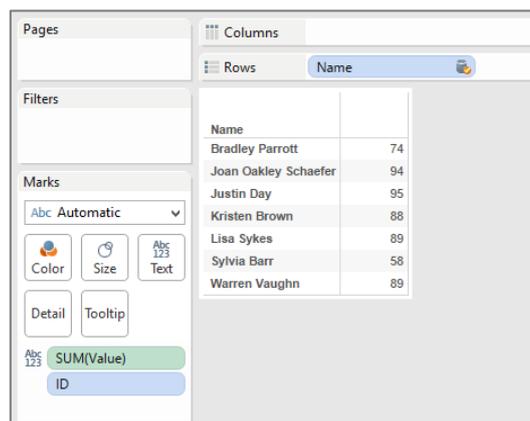
Diese Tabelle kann jetzt neu erstellt werden, ohne dass eine Verschmelzung nötig wäre.



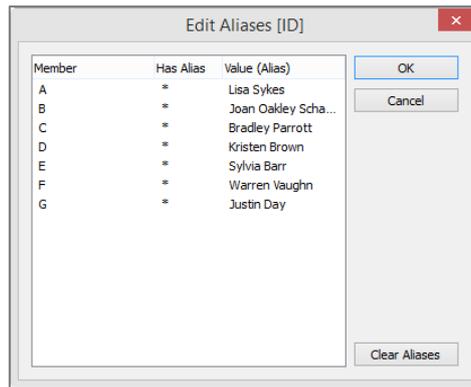
Primäraliase sind nützlich, wenn die sekundäre Datenquelle ein Attribut enthält, das 1:1 zurück zu den Dimensionselementen in der primären Datenquelle zugeordnet ist. Angenommen, aus den oben genannten Daten soll Folgendes erstellt werden:

Name	Value
Lisa Sykes	89
Joan Oakley Schaefer	94
Bradley Parrott	74
Kristen Brown	88
Sylvia Barr	58
Warren Vaughn	89
Justin Day	95

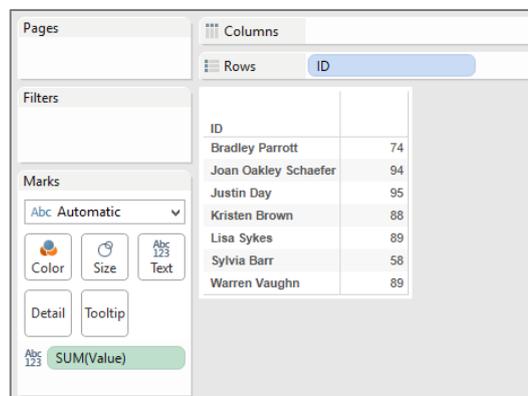
Dies kann durch Verschmelzung zwischen den beiden Datenquellen erfolgen. Aber wie bereits besprochen, würde dies zu einer sehr schwachen Leistung führen, wenn eine größere Anzahl von IDs vorhanden ist:



Wenn Sie auf das Feld „Name“ mit der rechten Maustaste klicken und „Primäre Aliase bearbeiten“ auswählen, können Sie eine einmalige Zuordnung des Namens- zum ID-Feld als Aliaswerte vornehmen:



Sie können jetzt die gewünschte Visualisierung ohne Verschmelzung erstellen. Dies wird viel schneller vonstatten gehen:



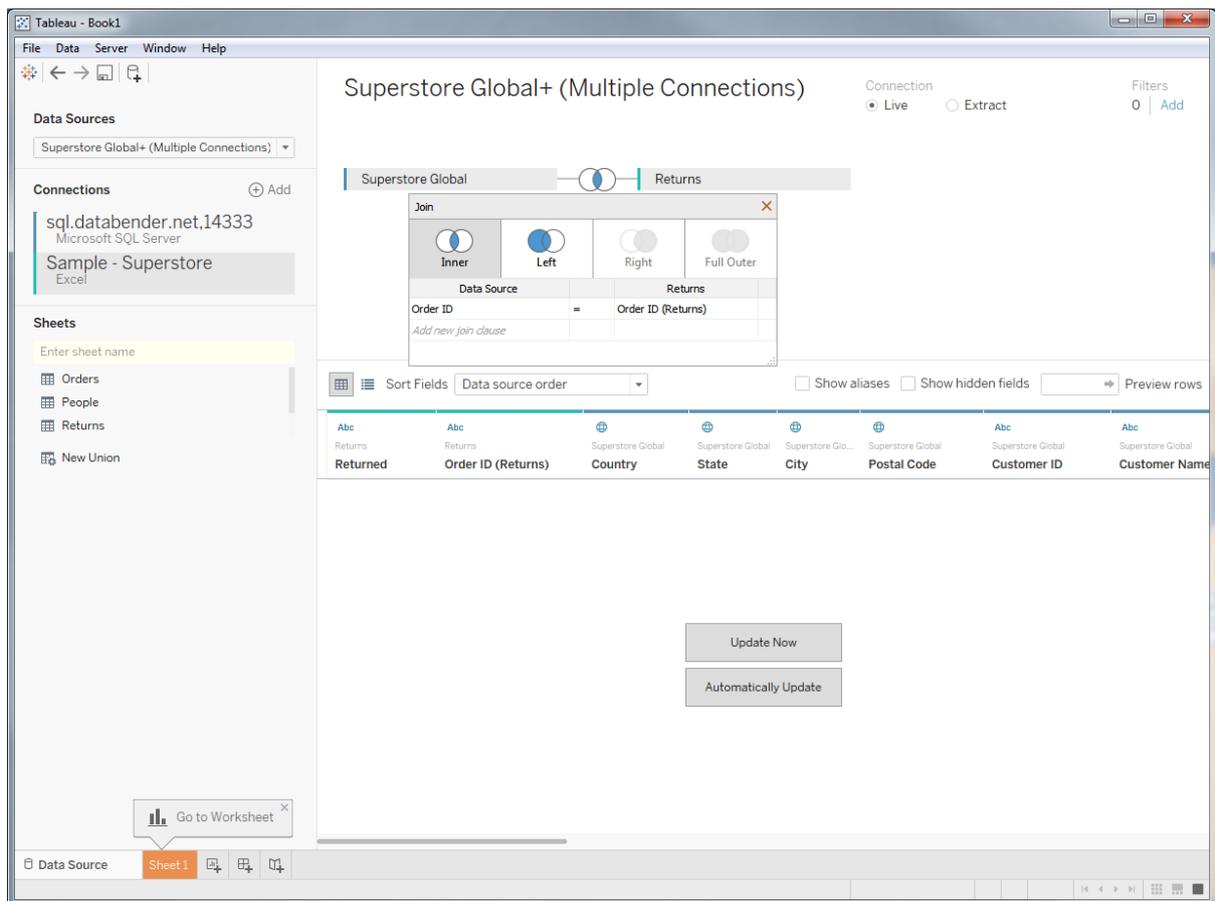
Beachten Sie, dass weder die primären Gruppen noch die primären Aliase dynamisch sind und aktualisiert werden müssen, wenn sich die Daten ändern. Daher ist dies keine sehr gute Lösung für häufig aktualisierte Daten. Wenn Sie hingegen eine schnelle Zuordnung benötigen, können Sie möglicherweise teure Verschmelzungen vermeiden.

Weitere Informationen finden Sie im folgenden Artikel in der Wissensdatenbank:

http://onlinehelp.tableau.com/current/pro/desktop/de-de/help.htm#multipleconnections_create_primary_group.html

Datenintegration

Die Datenintegration ist eine neue Funktion von Tableau 10, bei der in einer einzelnen Datenquelle Daten aus mehreren, potenziell heterogenen Datenverbindungen kombiniert werden können.



Ein wichtiger Unterschied zwischen Datenintegration und Datenverschmelzung ist, dass es sich bei der Datenintegration um eine Verknüpfung auf Zeilenebene handelt. Die Datenverschmelzung erfolgt anhand eines aggregierten Ergebnissatzes aus jeder einzelnen Datenquelle. Das bedeutet, dass die Datenintegration auf die Größe der zugrunde liegenden Datenquellen reagiert. Hierbei gibt es vier wichtige Überlegungen:

- Je mehr Daten bewegt werden müssen, desto länger dauert es: Daten werden aus jeder einzelnen Datenverbindung auf Zeilenebene extrahiert. Das Datenvolumen ist also ein Schlüsselfaktor.
- Je weiter die Daten reisen müssen, desto länger dauert es: Die Leistung wird beeinträchtigt, wenn Sie eine Verknüpfung zu einer Datenquelle herstellen, auf die über eine Verbindung mit hoher Latenz zugegriffen wird.
- Je niedriger der Datendurchsatz, desto länger dauert es: Die Leistung wird beeinträchtigt, wenn Sie eine Verknüpfung zu einer Datenquelle herstellen, auf die über eine Verbindung mit geringer Bandbreite zugegriffen wird.
- Je mehr Elemente übereinstimmen müssen, desto länger dauert es: Ähnlich wie unter Punkt 1 wird die Leistung von der Anzahl der Datensätze beeinträchtigt, die verknüpft werden sollen.

In Tableau 10 kann die Datenintegration nur mit extrahierten Datenquellen verwendet werden. Die Ergebnisse von jeder einzelnen Datenverbindung müssen also in eine TDE-Datei extrahiert werden. Diese extrahierte Datenquelle kann dann auf Tableau Server veröffentlicht werden, damit andere sie benutzen können.

In Tableau 10 ist es nicht möglich, Datenintegration auf Live-Datenquellen anzuwenden. Es ist auch nicht möglich, Datenintegration mit veröffentlichten Datenquellen durchzuführen. Dies sind Funktionen, die wir in zukünftigen Erweiterungen hinzufügen werden.

SQL (benutzerdefiniert)

Manchmal wollen neue Tableau-Benutzer Techniken aus der alten Welt auf ihre Arbeitsmappen anwenden, etwa Datenquellen mithilfe handgeschriebener SQL-Anweisungen zu erstellen. In vielen Fällen ist dies kontraproduktiv, weil Tableau Abfragen wesentlich effizienter generieren kann, wenn wir einfach die Verknüpfungsbeziehungen zwischen den Tabellen definieren und das Schreiben der SQL für die zu erstellende Ansicht der Abfrage-Engine überlassen. Es gibt jedoch Fälle, wenn das Spezifizieren von Verknüpfungen im Bereich „Datenverbindung“ nicht die gewünschte Flexibilität bietet, um die Beziehungen in Ihren Daten zu definieren.

Eine Datenverbindung mithilfe einer handgeschriebenen SQL-Anweisung zu schreiben, kann sehr leistungsstark sein, aber wie in den Spider-Man-Comics immer betont wird, kommt „mit großer Macht große Verantwortung“. In einigen Fällen kann benutzerdefinierte SQL tatsächlich die Leistung reduzieren. Der Grund dafür ist Folgender: Im Gegensatz zur Definition von Verknüpfungen wird benutzerdefinierte SQL niemals aufgeschlüsselt und wird immer atomisch ausgeführt. Es findet also keine Verknüpfungsauswahl statt und letztendlich können Situationen entstehen, wo die Datenbank aufgefordert wird, für nur eine Spalte die gesamte Abfrage zu verarbeiten, wie in diesem Fall:

```
SELECT SUM([TableauSQL].[Sales])
FROM (
    SELECT [OrdersFact].[Order ID] AS [Order ID],
           [OrdersFact].[Date ID] AS [Date ID],
           [OrdersFact].[Customer ID] AS [Customer ID],
           [OrdersFact].[Place ID] AS [Place ID],
           [OrdersFact].[Product ID] AS [Product ID],
           [OrdersFact].[Delivery ID] AS [Delivery ID],
           [OrdersFact].[Discount] AS [Discount],
           [OrdersFact].[Cost] AS [Cost],
           [OrdersFact].[Sales] AS [Sales],
           [OrdersFact].[Qty] AS [Qty],
           [OrdersFact].[Profit] AS [Profit]
    FROM [dbo].[OrdersFact] [OrdersFact]
) [TableauSQL]
HAVING (COUNT_BIG(1) > 0)
```

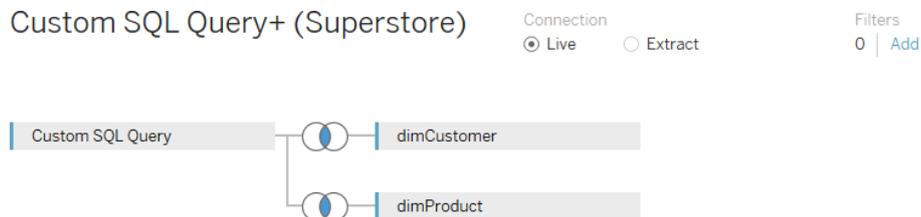
Es ist wichtig sicherzustellen, dass Ihre benutzerdefinierte SQL-Anweisung keine unnützen Klauseln enthält. Wenn Ihre Abfrage zum Beispiel GROUP BY- oder ORDER BY-Klauseln enthält, werden diese im Allgemeinen zu zusätzlichem Aufwand, weil Tableau auf der Grundlage der Struktur der Visualisierung eigene Klauseln erstellt. Eliminieren Sie diese aus Ihrer Abfrage, sofern möglich.

Eine gute Empfehlung ist es, benutzerdefinierte SQL in Verbindung mit Datenextrakten zu verwenden. Auf diese Weise wird die atomische Abfrage nur einmal ausgeführt (zum Laden der Daten in den Datenextrakt) und sämtliche nachfolgenden Analysen in Tableau werden mithilfe dynamischer, optimierter Abfragen im Datenextrakt ausgeführt. Selbstverständlich haben alle Regeln ihre Ausnahmen, so auch diese: Wenn Sie benutzerdefinierte SQL verwenden, erstellen Sie einen Datenextrakt, ES SEI DENN, Ihre benutzerdefinierte SQL enthält Parameter.

Parameter in einer benutzerdefinierten SQL-Anweisung zu verwenden, kann Direktverbindungen in einigen Fällen leistungsstärker machen, weil die zugrunde liegende Abfrage dynamischer wird (z. B. Filterklauseln mit Parametern werden angemessen evaluiert). Dies kann auch verwendet werden, um Werte für Leistungsbegrenzer zu übergeben, wie etwa TOP oder SAMPLE, um die von der Datenbank zurückgegebene Datenmenge einzuschränken. Wenn Sie jedoch einen Datenextrakt

verwenden, wird dieser jedes Mal, wenn Sie den Parameter ändern, geleert und neu generiert. Dies kann viel Zeit in Anspruch nehmen. Beachten Sie auch, dass Parameter nur verwendet werden können, um literale Werte zu übergeben. Sie können also nicht verwendet werden, um die SELECT- oder FROM-Klauseln dynamisch zu ändern.

Schließlich ist es auch möglich, Tabellen mit benutzerdefinierter SQL zu verknüpfen:



So können Sie eine spezifischere benutzerdefinierte SQL schreiben, die sich auf eine Teilmenge des Gesamtschemas bezieht. Die Verknüpfungen von dieser zu anderen Tabellen können dann möglicherweise wie normale Tabellenverknüpfungen ausgewählt werden, sodass effizientere Abfragen entstehen.

Alternativen zu benutzerdefinierter SQL

Anstatt benutzerdefinierte SQL direkt in Tableau zu verwenden, ist es manchmal vorzuziehen, die Abfrage in die zugrunde liegende Datenquelle zu verlagern. In vielen Fällen kann dies die Leistung verbessern, weil die Datenquelle die Query effizienter analysieren kann. Es kann aber auch bedeuten, dass eine komplexe Abfrage nur einmal ausgeführt zu werden braucht. Es eignet sich auch gut zu guten Managementverfahren, weil die einzelne Definition für mehrere Arbeitsmappen und Datenquellen freigegeben werden kann.

Es gibt verschiedene Möglichkeiten, dies zu erreichen:

Ansichten

Fast alle DBMS unterstützen das Konzept der Ansichten: Eine virtuelle Tabelle repräsentiert das Ergebnis einer Datenbankabfrage. Bei einigen Datenbanksystemen wird eine deutlich bessere Leistung erreicht, indem die Abfrage einfach aus einer benutzerdefinierten SQL-Anweisung herausgenommen und als Ansicht in der Datenbank instanziiert wird. Der Grund dafür ist, dass der Abfrageoptimierer in der Lage ist, einen besseren Ausführungsplan zu generieren, als wenn die Abfrage in einer äußeren SELECT-Anweisung verpackt ist. Detaillierte Informationen finden Sie in der folgenden Community-Diskussion:

<http://tabsoft.co/1su5YMe>

Das Definieren einer benutzerdefinierten Abfragelogik in einer Ansicht, anstatt in einer Tableau-Arbeitsmappe, bietet auch die Möglichkeit, sie in mehreren Arbeitsmappen und Datenquellen wiederzuverwenden. Außerdem unterstützen zahlreiche DBMS das Konzept materialisierter Ansichten oder Momentaufnahmen, wo die Ergebnisse der Ansichtsabfrage vorab berechnet und im Zwischenspeicher abgelegt werden, sodass während der Abfragezeit eine schnellere Reaktion erzielt wird. Sie ähneln möglicherweise Zusammenfassungstabellen (siehe unten), werden jedoch automatisch von der DBMS vorgehalten.

Gespeicherte Prozeduren

Gespeicherte Prozeduren ähneln den Ansichten, können jedoch eine viel komplexere Logik enthalten und mehrere Schritte der Datenaufbereitung ausführen. Sie können auch Parameter enthalten, sodass sie je nach Benutzereingabe einen gezielten Datenbestand zurückgeben können.

Gespeicherte Prozeduren werden in Sybase ASE, SQL Server und Teradata unterstützt. Um zu vermeiden, dass die gespeicherte Prozedur für nur eine Visualisierung mehrmals ausgeführt wird, führt Tableau Sie aus und schreibt den Ergebnissatz in eine temporäre Tabelle in der Datenbank. Die eigentlichen Abfragen für die Visualisierung werden dann in der temporären Tabelle ausgeführt. Die Ausführung der gespeicherten Prozedur und das Ausfüllen der temporären Tabelle erfolgt beim ersten Öffnen der Arbeitsmappe und immer, wenn die Parameter der gespeicherten Prozedur geändert werden. Dieser Prozess kann einige Zeit in Anspruch nehmen und diese Interaktionen sind möglicherweise langsam.

Wenn Sie die Ergebnisse einer gespeicherten Prozedur mit Parametern in einen Datenextrakt extrahieren, wird der Extrakt jedes Mal geleert und aktualisiert, wenn Sie die Parameterwerte ändern.

Weitere Informationen über den Umgang mit gespeicherten Prozeduren finden Sie in der Online-Hilfe von Tableau:

<http://tabsoft.co/1HAD93e>

Zusammenfassungstabellen

Wenn Sie sehr große, detaillierte Datenbestände haben, die Sie normalerweise beim Abfragen zusammenfassen (wenn Sie z. B. einzelne Transaktionen speichern, die Daten jedoch nach Tagen, Regionen, Kunden, Produkten usw. zusammenfassen), sollten Sie die Möglichkeit in Betracht ziehen, eine Zusammenfassungstabelle zu erstellen und Tableau mit dieser arbeiten zu lassen, damit die Abfragen schneller ausgeführt werden.

Hinweis: Sie können Tableau-Datenextrakte verwenden, um ein ähnliches Ergebnis zu erzielen, indem Sie einen aggregierten Datenextrakt erstellen. Detaillierte Hinweise finden Sie im Abschnitt über Extrakte.

SQL-Anfangsdaten

Eine Alternative für benutzerdefinierte SQL (falls Ihre Datenquelle dies unterstützt) besteht darin, die benutzerdefinierte SQL-Anweisung in einem SQL-Anfangsdaten-Block zu verwenden. Sie können dies einsetzen, um eine temporäre Tabelle zu erstellen, die anschließend in Ihrer Abfrage die ausgewählte Tabelle sein wird. Da die SQL-Anfangsdaten nur einmal ausgeführt werden, wenn die Arbeitsmappe geöffnet wird (und nicht jedes Mal, wenn die Visualisierung für benutzerdefinierte SQL geändert wird), kann dies die Leistung in einigen Fällen signifikant verbessern.

Beachten Sie, dass ein Administrator bei Tableau Server eine Einschränkung für SQL-Anfangsdaten festlegen kann, sodass sie nicht ausgeführt werden. Sie müssen prüfen, ob dies in Ihrer Umgebung akzeptabel ist, falls Sie Ihre Arbeitsmappe veröffentlichen und für andere freigeben möchten.

Weitere Informationen über SQL-Anfangsdaten finden Sie in der Online-Dokumentation:

http://onlinehelp.tableau.com/current/pro/desktop/de-de/help.htm#connect_basic_initialsql.html

Liegt es an meinen Daten?

Eine der leistungsstärksten Funktionen von Tableau ist die Fähigkeit, Daten aus verschiedenen Plattformen zu verbinden. Allgemein ausgedrückt, können diese Plattformen die folgenden Merkmale aufweisen:

- Dateibasierte Datenquellen: etwa Excel und CSV
- Relationale Datenbank-Datenquellen: etwa Oracle, Teradata und SQL Server sowie spezialisierte analytische Geräte wie HP Vertica, IBM Netezza usw.
- OLAP-Datenquellen: wie Microsoft Analysis Services und Oracle Essbase
- Big Data-Datenquellen: etwa Hadoop
- Cloudbasierte Datenquellen: etwa Salesforce, Google usw.

Jeder Datenquellentyp hat seine eigenen Vor- und Nachteile und wird auf spezifische Weise behandelt.

Beachten Sie, dass Tableau Desktop sowohl auf Windows als auch auf Mac OS X unterstützt wird. Die unterstützten Datenquellen sind unter Windows und Mac OS jedoch nicht identisch. Tableau arbeitet daran, die Unterschiede zwischen den Plattformen zu minimieren, aber derzeit gibt es einige Datenquellen, die nur auf einer der Plattformen unterstützt werden.

Allgemeiner Ratschlag

Native Treiber verwenden

Tableau 10 unterstützt native Verbindungen zu mehr als 40 verschiedenen Datenquellen. Das bedeutet, dass Tableau Techniken, Fähigkeiten und Optimierungen speziell für diese Datenquellen implementiert hat. Konzeptions- und Testaktivitäten für diese Verbindungen gewährleisten, dass sie die robustesten sind, die Tableau zu bieten hat.

Tableau unterstützt auch allgemeine ODBC für den Zugriff auf Datenquellen außerhalb der Liste der nativen Datenkonnektoren. Für diesen öffentlich definierten Standard stellen viele Datenbankanbieter ODBC-Treiber bereit und Tableau kann auch diese ODBC-Treiber verwenden um eine Verbindung zu den Daten herzustellen. Datenbankanbieter können die Funktionen des ODBC-Standards auf unterschiedliche Weise interpretieren oder implementieren. In einigen Fällen empfiehlt oder benötigt Tableau einen Datenextrakt, um mit einem bestimmten Treiber arbeiten zu können. Es gibt auch einige ODBC-Treiber und Datenbanken, zu denen Tableau keine Verbindung herstellen kann.

Falls ein nativer Treiber für die von Ihnen abgefragte Datenquelle vorhanden ist, sollten Sie lieber diesen als die ODBC-Verbindungen verwenden, weil Sie im Allgemeinen eine bessere Leistung erzielen. Beachten Sie auch, dass ODBC-Verbindungen nur unter Windows verfügbar sind.

Testen Sie so nah wie möglich an den Daten

Wie weiter oben festgestellt gilt ein allgemeines Prinzip: Wenn eine Datenquelle Abfragen langsam ausführt, ist die Nutzererfahrung in Tableau langsam. Eine gute Möglichkeit, die Rohleistung einer Datenquelle zu testen, besteht darin, sofern möglich, Tableau Desktop auf der Maschine zu installieren, wo sich die Datenquelle befindet, und einige Abfragen auszuführen. Auf diese Weise werden Faktoren wie Netzwerkbandbreite und Latenz aus der Leistungsbeurteilung eliminiert. So können Sie die Rohleistung der Abfrage in der Datenquelle besser verstehen. Wenn Sie außerdem den Namen *localhost* für die Datenquelle verwenden, anstatt den DNS-Namen, kann dies dazu beitragen festzustellen, ob Umgebungsfaktoren, wie etwa eine langsame Namensauflösung oder Proxy-Server zur schwachen Leistung beitragen.

Datenquellen

Dateien

Diese Kategorie deckt datebasierte Datenformate ab: Textdateien wie CSV, Excel-Spreadsheets und MS Access sind die häufigsten. Es gehören jedoch auch Datendateien von statistischen Plattformen wie SPSS, SAS und R dazu. Geschäftsanwender verwenden häufig Daten in diesen Formaten, weil es sich um eine gängige Art handelt, Daten aus „regulierten“ Datenbeständen abzurufen, entweder mithilfe von Berichten oder einem Abfrageextrakt.

Im Allgemeinen wird empfohlen, datebasierte Datenquellen in die schnelle Tableau-Datenengine zu importieren. Dadurch werden die Abfragen schneller ausgeführt und die Datei zum Speichern der Datenwerte wird kleiner. Wenn die Datei allerdings klein ist oder wenn Sie eine Direktverbindung zur Datei benötigen, damit sich verändernde Daten sichtbar werden, können Sie eine direkte Verbindung herstellen.

Schattenextrakte

Wenn Sie eine Verbindung zu nicht veralteten Excel-/Text- oder Statistikdateien herstellen, erstellt Tableau beim Herstellen der Verbindung auf transparente Weise eine Extraktdatei. Dies wird Schattenextrakt genannt. Dadurch wird die Arbeit mit den Daten viel schneller als mit einer Direktabfrage in der Datei.

Möglicherweise beobachten Sie, dass bei der ersten Benutzung einer großen Datei mehrere Sekunden vergehen, bis die Daten im Vorschaubereich erscheinen. Der Grund dafür ist, dass Tableau die Daten aus der Datei extrahiert und in

`C:\Users\\AppData\Local\Tableau\Caching\TemporaryExtracts` eine Schattenextraktdatei schreibt. Standardmäßig wird diese Datei in `extracts` erstellt, und zwar mit einem Hash-Namen aus dem Pfad und dem letzten Änderungsdatum der Datendatei. Tableau behält Schattenextrakte für die fünf zuletzt benutzten Dateidatenquellen in diesem Verzeichnis. Die älteste Datei wird gelöscht, wenn eine neue erstellt wird. Wenn Sie anschließend eine Datei mit Schattenextrakt erneut benutzen, öffnet Tableau einfach die Extraktdatei und die Datenvorschau erscheint verzögerungsfrei.

Obwohl die Schattenextraktdatei zugrunde liegenden Daten und andere Informationen enthält, ähnlich wie der standardmäßige Tableau-Extrakt, werden Schattenextraktdateien in einem anderen Format gespeichert (mit der Erweiterung `.ttde`). Das bedeutet, dass sie nicht auf die gleiche Weise wie Tableau-Extrakte genutzt werden können.

Datenkonnektoren für veraltete Excel- und Textdateiverbindungen

In früheren Versionen von Tableau nutzten Verbindungen zu Excel- und Textdateien den Datenengine-Treiber JET von Microsoft. Seit Tableau 8.2 wird standardmäßig ein nativer Treiber verwendet, der eine bessere Leistung erzielt und mit größeren, komplexeren Dateien arbeiten kann. Es gibt jedoch einige Situationen, in denen veraltete Treiber besser sein können: zum Beispiel, wenn Sie Nutzer definierte SQL verwenden. In diesen Szenarios haben Benutzer die Option, zum veralteten JET-Treiber zurückzuwechseln. Beachten Sie, dass für das Lesen von MS Access-Dateien weiterhin der JET-Treiber erforderlich ist.

Eine detaillierte Liste der Unterschiede zwischen den beiden Treibern finden Sie hier:

http://onlinehelp.tableau.com/current/pro/desktop/de-de/help.htm#upgrading_connection.html

Beachten Sie, dass die JET-Treiber auf Mac OS mit verfügbar sind und Tableau Desktop für Mac deshalb keine MS Access-Dateien unterstützt. Auch die veraltete Verbindungsoption für Excel- und Textdateien ist nicht verfügbar.

Relationale Datenbanken

Relationale Datenquellen sind die häufigste Datenquellenform für Tableau-Benutzer und Tableau bietet native Treiber für eine breite Palette von Plattformen. Diese können zeilen- oder spaltenbasiert, personen- oder unternehmensbezogen sein und über native Treiber bzw. generisches ODBC einbezogen werden. Diese Kategorie umfasst technisch gesehen MapReduce-Datenquellen, weil diese über SQL-Zugriffsebenen wie Hive oder Impala einbezogen werden, jedoch werden wir weiter unten im Abschnitt „Big Data“ näher darauf eingehen.

Es gibt viele interne Faktoren, die die Abfragegeschwindigkeit in einem relationalen Datenbanksystem (RDBMS) beeinflussen. Die Änderung oder Feinabstimmung dieser Datenbanken erfordert in der Regel die Unterstützung durch Ihren DBA, kann jedoch signifikante Leistungsverbesserungen erbringen.

Zeilenbasiert vs. spaltenbasiert

RDBMS gibt es vor allem in zwei Geschmacksrichtungen: zeilenbasiert oder spaltenbasiert. Zeilenbasierte Speicherlayouts eignen sich gut für OLTP-ähnliche Arbeitslasten mit vielen interaktiven Transaktionen. Spaltenbasierte Speicherlayouts eignen sich gut für analytische Arbeitslasten (z. B. Data Warehouses), bei denen häufig hochkomplexe Abfragen in großen Datenbeständen ausgeführt werden.

Heute basieren viele Hochleistung-Analyselösungen auf spaltenbasierten RDBMS und Sie werden feststellen, dass Ihre Abfragen schneller ausgeführt werden, wenn Sie solch eine Lösung verwenden. Beispiele für von Tableau unterstützte spaltenbasierte Datenbanken sind Actian Vector, Amazon Redshift, HP Vertica, IBM Netezza, MonetDB, Pivotal Greenplum, SAP HANA und SAP Sybase IQ.

SQL als Schnittstelle

Es gibt viele Systeme, die nicht auf herkömmlichen RDBMS-Technologien basieren, aber dennoch als relationale Quelle erscheinen, weil sie eine SQL-basierte Schnittstelle anbieten. Für Tableau umfasst dies mehrere NoSQL-Plattformen (z. B. MarkLogic, DataStax, MongoDB usw.) sowie Plattformen zur Abfragebeschleunigung wie Kognitio, AtScale und JethroData.

Indizes

Die korrekte Indizierung Ihrer Datenbank ist wesentlich für eine gute Abfrageleistung:

- Vergewissern Sie sich, dass für alle an Tabellenverknüpfungen beteiligten Spalten Indizes vorhanden sind.
- Vergewissern Sie sich, dass für alle in Filtern verwendeten Spalten Indizes vorhanden sind.
- Berücksichtigen Sie, dass die Benutzung diskreter Datumsfilter in einigen Datenbanken zur Folge haben kann, dass Abfragen die Indizes von Date- und DateTime-Spalten nicht verwenden. Im Abschnitt über Filter wird noch näher darauf eingegangen: Durch einen Datumsbereichsfilter wird sichergestellt, dass der Datumsindex verwendet wird. Anstatt beispielsweise zu verwenden, drücken `YEAR([DateDim])=2010` Sie den Filter als `aus [DateDim] >= #2010-01-01# and [DateDim] <= #2010-12-31#`.
- Vergewissern Sie sich, dass die Statistiken für Ihre Daten aktiviert sind, damit der Abfrageoptimierer Abfragepläne von hoher Qualität erstellen kann.

Viele DBMS-Umgebungen verwenden Management-Tools, die eine Abfrage untersuchen und hilfreiche Indizes empfehlen.

NULL-WERTE

NULL-Werte in Dimensionsspalten können die Effektivität von Abfragen reduzieren. Ziehen Sie eine Visualisierung in Betracht, in der der Gesamtumsatz nach Land für die obersten 10 Produkte gezeigt werden soll. Tableau generiert zunächst eine Unterabfrage, um die obersten 10 Produkte zu finden, und diese dann mit der Abfrage nach Land zu verknüpfen, damit das Endergebnis ermittelt wird.

Wenn die Produktspalte auf ALLOW NULL gesetzt ist, müssen wir eine Abfrage ausführen, um zu prüfen, ob NULL von der Unterabfrage als eines der obersten 10 Produkte zurückgegeben wird. Wenn dies der Fall ist, müssen wir eine NULL-bewahrende Verknüpfung einrichten, die viel mehr Rechenaufwand erfordert als eine normale Verknüpfung. Wenn die Produktspalte auf NOT NULL gesetzt ist, wissen wir, dass das Ergebnis der Unterabfrage keine NULL enthalten kann. Wir können also eine normale Verknüpfung ausführen und die Abfrage „auf NULL prüfen“ überspringen.

Aus diesem Grunde sollten Sie, sofern möglich, Ihre Dimensionsspalten als NOT NULL definieren.

Referenzielle Integrität

Wenn Sie mehrere Tabellen in einer Datenquelle verknüpfen, verfügt Tableau über eine elegante (und für den Benutzer im Allgemeinen nicht sichtbare) Funktion namens „Verknüpfungsauswahl“. Da Verknüpfungen Aufwand an Zeit und Ressourcen auf dem Datenbankserver bedeuten, sollte nicht jedes Mal jede Verknüpfung aufgezählt werden, die in Ihrer Datenquelle deklariert wurde. Mit der Verknüpfungsauswahl besteht die Möglichkeit, nur die relevanten Tabellen abzufragen, anstatt alle in der Verknüpfung definierten Tabellen.

Sehen Sie sich das folgende Szenario an, in dem mehrere Tabellen in einem kleinen Sternschema verknüpft wurden:

#	Abc	#	Abc	#	#	Abc
CustomerDim	CustomerDim	DeliveryDim	DeliveryDim	DeliveryDim	LocationDim	LocationDim
Customer ID (Cust...	Name	Delivery ID (Delive...	Shipping Mode	Days	Place ID (Location...	Region
75.00	Altd	8900.00	First Class	3.00000	1051.00	Europe
1195.00	Systed Systems, Inc	8300.00	First Class	2.00000	661.00	Asia
1004.00	Seas	2400.00	Standard Class	5.00000	885.00	Asia
1439.00	Vellutems Corporation	0.00	Standard Class	4.00000	595.00	Asia
883.00	Mics Consulting	0.00	Standard Class	4.00000	786.00	Asia
984.00	Quan	2400.00	Standard Class	5.00000	173.00	Africa
1029.00	Sems Group	4200.00	Standard Class	6.00000	31.00	Africa
953.00	Nettpaq USA	2400.00	Standard Class	5.00000	130.00	Africa
683.00	Inetems Research	6000.00	Second Class	2.00000	282.00	America
1302.00	TER	5400.00	Standard Class	7.00000	541.00	Asia
1054.00	Sillian Corporation	0.00	Standard Class	4.00000	151.00	Africa

Wenn Sie bei aktivierter Verknüpfungsauswahl auf die Kennzahl „Umsatz“ doppelklicken, wird die folgende Abfrage generiert:

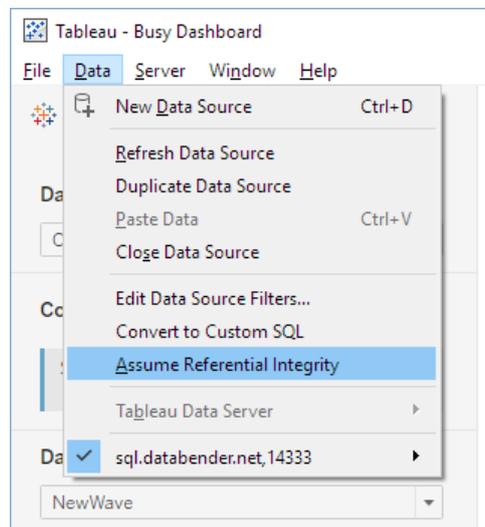
```
SELECT SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
GROUP BY ()
```

Ohne diese Option wird eine wesentlich weniger effiziente Abfrage generiert:

```
SELECT SUM([OrdersFact].[Sales]) AS [sum:Sales:ok]
FROM [dbo].[OrdersFact] [OrdersFact]
INNER JOIN [dbo].[CustomerDim] [CustomerDim]
ON ([OrdersFact].[Customer ID] = [CustomerDim].[Customer ID])
INNER JOIN [dbo].[DeliveryDim] [DeliveryDim]
ON ([OrdersFact].[Delivery ID] = [DeliveryDim].[Delivery ID])
INNER JOIN [dbo].[LocationDim] [LocationDim]
ON ([OrdersFact].[Place ID] = [LocationDim].[Place ID])
INNER JOIN [dbo].[ProductDim] [ProductDim]
ON ([OrdersFact].[Product ID] = [ProductDim].[Product ID])
INNER JOIN [dbo].[TimeDim] [TimeDim]
ON ([OrdersFact].[Date ID] = [TimeDim].[Date ID])
GROUP BY ()
```

Alle Dimensionstabellen müssen verknüpft sein, um sicherzustellen, dass von Beginn an korrekte Kennzahlensummen berechnet werden. Wenn Ihre Tabelle „Fakten“ beispielsweise Daten für 2008-2012 enthielt, aber die Tabelle „Zeitdimension“ nur Werte für 2010-2012 hatte, würde sich das Ergebnis SUM([Umsatz]) möglicherweise ändern, wenn die Tabelle „Zeit“ eingeschlossen wäre.

Bisher war „harte“ referenzielle Integrität erforderlich, wenn die Regel vom DBMS erzwungen wurde. Viele Kunden haben jedoch Datenquellen, bei denen die referenzielle Integrität entweder auf der Anwendungsschicht oder über einen ETL-Prozess erzwungen wird. Dies wird auch als „weiche“ referenzielle Integrität bezeichnet. Benutzer können Tableau mitteilen, dass weiche referenzielle Integrität festgelegt ist und dass die Verknüpfungsauswahl sicher verwendet werden kann, indem die Option „Referenzielle Integrität voraussetzen“ aktiviert wird.



Beachten Sie Folgendes: Tableau kann zwar harte oder weiche referenzielle Integrität verwenden, jedoch ist harte referenzielle Integrität häufig günstiger, weil die Datenbank die Verknüpfungsauswahl nutzen kann. Weitere Hinweise finden Sie in der folgenden Artikelserie von Russell Christopher auf seinem Tableau Love-Blog:

<http://bit.ly/1HACmPV>

<http://bit.ly/1HACqPn>

Partitionierung

Die Partitionierung einer Datenbank verbessert deren Leistung, weil eine große Tabelle in kleinere, individuelle Tabellen (genannt „Partitionen“ oder „Shards“) aufgeteilt wird. Das bedeutet, dass Abfragen schneller ablaufen können, weil weniger Daten zu durchsuchen sind und/oder mehr Laufwerke zur Bedienung der E/A-Vorgänge verfügbar sind. Die Partitionierung ist eine empfohlene Strategie für große Datenvolumen und sie ist für Tableau transparent.

Die Partitionierung funktioniert gut in Tableau, wenn sie über eine Dimension durchgeführt wird – zum Beispiel Zeit, Region, Kategorie usw. –, nach der häufig gefiltert wird, sodass einzelne Abfragen nur Datensätze innerhalb einer Partition zu lesen brauchen.

Achten Sie darauf, dass bei einigen Datenbanken Datumsbereichsfilter (keine diskreten Filter) erforderlich sind, um sicherzustellen, dass die Partitionsindizes korrekt verwendet werden. Andernfalls wird ein vollständiger Tabellenscan mit extrem schlechter Leistung durchgeführt.

Temporäre Tabellen

In Tableau gibt es zahlreiche Vorgänge, die zur Nutzung von temporären Tabellen führen können: beispielsweise Ad-hoc-Gruppen und Sätze erstellen und Daten verschmelzen. Es wird empfohlen, Ihren Benutzern die Berechtigung zum Erstellen und Verwerfen temporärer Tabellen zu erteilen und sicherzustellen, dass Ihre Umgebung über ausreichend Spool-Platz für die Abfragen verfügt, die sie ausführen.

OLAP

Tableau unterstützt mehrere OLAP-Datenquellen:

- Microsoft Analysis Services
- Microsoft PowerPivot (sowohl PowerPivot für Excel als auch PowerPivot für SharePoint)
- Oracle Essbase
- SAP Business Warehouse
- Teradata OLAP

Aufgrund der Sprachunterschiede zwischen MDX/DAX und SQL gibt es funktionelle Unterschiede je nachdem, ob Sie eine Verbindung zu OLAP- oder zu relationalen Datenquellen herstellen. Die zu berücksichtigenden Kernpunkte sind, dass beide in Tableau die gleiche Benutzeroberfläche, die gleichen Visualisierungen und die gleiche Ausdruckssprache für berechnete Kennzahlen haben. Die Unterschiede beziehen sich hauptsächlich auf Metadaten (wie und wo sie definiert werden), auf Filterung, wie Summen und Aggregationen funktionieren und wie die Datenquelle bei der Datenverschmelzung verwendet werden kann.

Weitere Informationen zu den Unterschieden bei der Nutzung von Tableau mit relationalen vs. OLAP-Datenquellen finden Sie im folgenden Artikel in der Wissensdatenbank:

<http://tabsoft.co/1HAF47P>

SAP BW-Datenextrakte

SAP BW ist ein Sonderfall bei unseren OLAP-Datenquellen, weil Sie Daten aus SAP BW-Cubes in die Daten-Engine von Tableau extrahieren können (Hinweis: Dies erfordert einen speziellen Schlüsselcode, der von Tableau bezogen werden kann). Tableau ruft Blattebenenknoten ab (keine Daten von Drill-Through-Ebenen) und wandelt sie in eine relationale Datenquelle um. Da bei der Umwandlung einer multidimensionalen in eine relationale Datenquelle nicht alle Würfelstrukturen beibehalten werden, wird das freie Hin- und Zurückwechseln zwischen Extrakt und Direktverbindung ohne Einfluss auf den Zustand Ihrer Visualisierung für Cube-Extrakte nicht unterstützt. Sie müssen eine Wahl treffen, bevor Sie Ihre Visualisierung aufbauen. Sie brauchen allerdings nicht alles im Voraus zu entscheiden. Nach der Extraktion können Sie zwischen Aliasoptionen (Schlüssel, Langname usw.) wechseln.

Detaillierte Informationen zum Erstellen von SAP BW-Extrakten finden Sie unter:

<http://tabsoft.co/1SuYf9d>

Big Data

Big Data ist ein äußerst überladener Begriff in der Welt der Datenanalysen. In diesem Dokument verwenden wir ihn insbesondere für Plattformen auf Hadoop-Basis. In Tableau 10 gibt es vier unterstützte Hadoop-Distributionen, die Hive- und/oder Impala-Verbindungen unterstützen.

- Amazon EMR
 - HiveServer
 - HiveServer2
 - Impala
- Cloudera Hadoop
 - HiveServer
 - HiveServer2
 - Impala

- Hortonworks Hadoop Hive
 - HiveServer
 - HiveServer2
 - Hortonworks Hadoop Hive
- MapR Hadoop Hive
 - HiveServer
 - HiveServer2
- Spark SQL
 - SharkServer *
 - SharkServer2 *
 - SparkThriftServer

** Beachten Sie, dass SharkServer- und SharkServer2-Verbindungen für Sie bereitgestellt werden, Tableau jedoch keinen Support dafür leistet.*

Hive agiert als SQL-Hadoop- Übersetzungsebene, die die Abfrage in MapReduce übersetzt, welche anschließend mit HDFS-Daten ausgeführt wird. Impala führt die SQL-Anweisung direkt mit HDFS-Daten aus (unter Umgehung von MapReduce). Tableau unterstützt auch Spark SQL, eine Open Source-Verarbeitungs-Engine für Big Data, die bis zu 100 Mal schneller als MapReduce ausführen kann, weil sie speicherresident ausgeführt wird, statt auf der Festplatte.

Impala ist im Allgemeinen viel schneller als Hive, und Spark ist sogar noch schneller.

Auch mit diesen zusätzlichen Komponenten reagiert Hadoop häufig nicht ausreichend schnell für analytische Abfragen, etwa wie die von Tableau erstellten. Tableau-Datenextraktionen werden häufig verwendet, um die Reaktionszeiten von Abfragen zu verbessern. Weitere Hinweise zu Extrakten und wie sie mit „Big Data“ eingesetzt werden können, werden weiter unten diskutiert.

Detaillierte Hinweise zur Verbesserung der Leistung bei Verwendung von Hadoop-Datenquellen finden Sie hier:

<http://kb.tableau.com/articles/knowledgebase/hadoop-hive-performance?lang=de-de>

Cloud

Tableau unterstützt derzeit die folgenden Cloud-Datenquellen:

- Salesforce.com
- Google Analytics
- oData (einschließlich Windows Azure Marketplace DataMarket)

Diese erste Gruppe von Datenquellen liest eine Teilmenge von Datensätzen aus einem Webdienst und lädt sie in eine Tableau-Datenextraktionsdatei. „Direktverbindung“ ist keine Option für diese Datenquellen. Die Extraktdatei und die darin enthaltenen Daten können jedoch aktualisiert werden. Mit Tableau Server kann dieser Aktualisierungsprozess automatisiert und terminiert werden.

Tableau unterstützt auch die folgenden cloudbasierten Datenplattformen:

- Amazon Aurora
- Amazon Redshift
- Amazon RDS
- Google BigQuery

- Google Cloud SQL
- Google Tabellen
- Microsoft Azure SQL Data Warehouse
- Snowflake

Im Gegensatz zur vorherigen Datenquellengruppe funktionieren diese wie relationale Datenquellen und ermöglichen Direktverbindungen und Extrakte. Wir werden hier nicht näher darauf eingehen (siehe Abschnitt über relationale Datenquellen weiter oben). Wir möchten lediglich hervorheben, dass Sie sie im Allgemeinen mit Direktverbindung verwenden sollten, um zu vermeiden, dass große Datenvolumen aus der Cloud übertragen werden.

Schließlich bietet Tableau auch einen generischen Datenkonnektor zum Importieren von Daten von webbasierten Diensten, die Daten in JSON, XML oder HTML-Formaten veröffentlichen – den Web-Datenkonnektor.

Salesforce

Wenn Sie eine Verbindung zu Salesforce herstellen, müssen die folgenden Einschränkungen in Betracht gezogen werden:

Es gibt keine Option „Direktverbindung“

Es gibt mehrere Gründe dafür, dass wir auf Extrakte setzen, anstatt auf Direktverbindungen:

- Leistung – Live-Analyseabfragen in Salesforce sind (im Allgemeinen) langsam.
- API-Quoten – Wenn Salesforce zu häufig direkt angesteuert wird, kann das Konto suspendiert werden, sofern die Tagesquote erreicht wurde. Durch die Extraktion nutzen wir die APIs auf effiziente Weise, um die Anzahl der erforderlichen API-Aufrufe zu minimieren und zu vermeiden, dass Grenzwerte erreicht werden. Um die optimale Leistung aufrechtzuerhalten und sicherzustellen, dass die Force.com-API immer für alle ihre Kunden erreichbar ist, gleicht Salesforce.com die Transaktionslasten aus, in dem es zwei Arten von Grenzwerten durchsetzt: Concurrent API Request Limits (<http://sforce.co/1f19cQa>) und Total API Request Limits (<http://sforce.co/1f19kiH>).

Der Anfangsextrakt kann sehr langsam sein

Das erste Mal, wenn Sie die Daten aus Salesforce extrahieren, kann es je nach Tabellengröße, force.com-Auslastung usw. eine Weile dauern. Der Grund dafür ist, dass Objekte vollständig heruntergeladen werden.

Verknüpfungsoptionen sind begrenzt

Wenn Sie die Option „Mehrere Tabellen“ wählen, bedenken Sie, dass Sie Objekte nur über ihre PK/FK-Schlüssel verknüpfen können (nur Left Join und Inner Join).

Daten können nicht vorab gefiltert werden

Es gibt keine Möglichkeit, die Daten über den Salesforce-Datenkonnektor vorab zu filtern. Wenn dies für sie kritisch ist, könnten Sie einen Salesforce-ODBC-Treiber eines externen Anbieters verwenden (zum Beispiel von Simba oder DataDirect), der Direktverbindungen unterstützt. Dann können Sie einen Extrakt über diese Verbindung erstellen.

DBAmp bietet auch eine Lösung, mit der Sie Salesforce-Daten in eine SQL-Server-Datenbank importieren können. Anschließend können Sie über den SQL Server-Datenkonnektor mit Tableau verbinden.

Formelspalten können nicht migriert werden

Wenn Sie berechnete Felder haben, müssen Sie sie in Tableau neu erstellen, nachdem Sie die Daten extrahiert haben.

Abfragen sind auf 10.000 Zeichen begrenzt

Die Force.com API begrenzt Abfragen auf insgesamt 10.000 Zeichen. Wenn Sie eine Verbindung zu einer oder mehreren Tabellen herstellen, die sehr breit sind (viele Spalten mit potenziell langen Spaltennamen), können Sie dieses Limit erreichen, wenn Sie versuchen, einen Extrakt zu erstellen. In diesen Fällen sollten Sie weniger Spalten auswählen, damit die Größe der Abfrage reduziert wird. In einigen Fällen kann Salesforce.com in der Lage sein, dieses Abfragelimit für Ihr Unternehmen zu erhöhen. Weitere Informationen erhalten Sie von Ihrem Salesforce-Administrator.

Google Analytics

Google Analytics (GA) entnimmt Ihren Daten Stichproben, wenn ein Bericht eine große Anzahl Dimensionen oder Daten enthält. Wenn Ihre Daten für eine bestimmte Webeigenschaft innerhalb eines vorgegebenen Datumsbereichs 50.000 Besuche überschreitet (bei einem regulären GA-Account), aggregiert GA die Ergebnisse und gibt eine Stichprobe dieser Daten zurück. Wenn GA eine Stichprobe dieser Daten an Tableau zurückgegeben hat, zeigt Tableau rechts unten in der betreffenden Ansicht die folgende Meldung an:

„Google Analytics hat Stichprobendaten zurückgegeben. Stichproben werden erstellt, wenn die Verbindung eine große Anzahl Dimensionen oder Daten enthält. Weitere Informationen zur Auswirkung von Stichproben auf Ihre Berichtsergebnisse finden Sie in der Google Analytics-Dokumentation.“

Es ist wichtig zu wissen, wann die Stichproben aus Ihren Daten entnommen werden, weil die Aggregation bestimmter Stichprobendaten stark verzerrte und ungenaue Schlussfolgerungen verursachen kann. Angenommen Sie aggregieren Stichprobendaten aus einer ungewöhnlichen Kategorie Ihrer Daten. Schlussfolgerungen aus den aggregierten Stichprobendaten können verzerrt sein, weil die betreffende Kategorie nur eine unzureichende Anzahl von Stichproben enthält. Zum Aufbau von GA-Ansichten, aus denen Sie präzise Schlüsse über Ihre Daten ziehen können, müssen Sie sich vergewissern, dass die zu beurteilende Kategorie ausreichend viele Daten enthält. Die empfohlene Mindestgröße für Stichproben ist 30.

Informationen über die Anpassung der GA-Stichprobengröße und weitere Hinweise zur GA-Stichprobennahme finden Sie in der GA-Dokumentation:

<https://support.google.com/analytics/answer/1042498?hl=de>

Zur Vermeidung der Stichprobennahme gibt es zwei Ansätze:

- Führen Sie mehrere GA-Berichte auf Sitzungs- oder Trefferebene aus, um die Daten in Teilmengen ohne Stichproben aufzuteilen. Als Nächstes laden Sie die Daten in eine Excel-Datei herunter und verwenden die Extrakt-Engine von Tableau, um Daten aus einer Datenquelle hinzuzufügen und die Daten wieder in einem Datensatz zusammenzufügen.
- Abonnieren Sie eine GA-Premium-Account. Damit wird die Anzahl der Datensätze erhöht, die in einen Bericht einbezogen werden können. Auf diese Weise ist es viel einfacher, die Daten für die Analyse aufzuteilen. Für die Zukunft hat Google angekündigt, dass GA-Premium-Kunden Ihre Daten von der Sitzungs- und Trefferebene in Google BigQuery exportieren können, um sie weiter zu analysieren. Dieser Ansatz wäre sehr viel bequemer, weil Tableau eine Direktverbindung zu BigQuery herstellen kann.

Beachten Sie schließlich, dass die API, die Tableau benutzt, um GA abzufragen, die Abfrage auf ein Maximum von sieben Dimensionen und zehn Kennzahlen beschränkt.

Web-Datenkonnektor

Ein Tableau-Web-Datenkonnektor bietet Ihnen die Möglichkeit, eine Verbindung zu Daten herzustellen, für die noch kein Datenkonnektor vorhanden ist. Mit einem Web-Datenkonnektor können Sie eine Verbindung zu fast allen Daten herstellen, auf die man über HTTP zugreifen kann. Dazu gehören interne Web-Services, JSON-Daten, XML-Daten, REST APIs und viele weitere Quellen. Da Sie kontrollieren, wie die Daten abgerufen werden, können Sie auch Daten aus mehreren Quellen kombinieren.

Einen Web-Datenkonnektor erstellen Sie, indem Sie eine Webseite schreiben, die JavaScript und HTML enthält. Nachdem Sie einen Web-Datenkonnektor geschrieben haben, können Sie ihn für andere Tableau-Benutzer freigeben, indem Sie ihn auf Tableau Server veröffentlichen.

Um Ihnen beim Erstellen von Web-Datenkonnektoren zu helfen, haben wir ein Softwareentwicklungs-Kit (SDK) konzipiert, das Vorlagen, Beispielcode und einen Simulator enthält, mit dem Sie Web-Datenkonnektoren testen können. Diese Dokumentation umfasst auch ein Lernprogramm, das Ihnen schrittweise zeigt, wie ein Web-Datenkonnektor erstellt wird. Das Kit für Web-Datenkonnektoren ist ein Open-Source-Projekt. Weitere Informationen finden Sie auf der Seite „Tableau webdatenconnector“ auf GitHub.

Weitere Informationen über Web-Datenkonnektoren finden Sie hier:

<http://tabsoft.co/1NnGsBU>

Datenvorbereitung

Manchmal sind die Daten, die wir in unseren Arbeitsmappen verwenden müssen, nicht perfekt. Möglicherweise sind sie nicht bereinigt, eventuell nicht in der richtigen Form und über mehrere Dateien oder Datenbanken verteilt. Bisher arbeitete Tableau am besten, wenn es bereinigte, normalisierte Daten bekam. Das bedeutete, dass Sie die Daten gelegentlich mithilfe anderer Tools aufbereiten mussten, bevor Sie sie in Tableau laden konnten.

Dies trifft nicht mehr immer zu: Tableau verfügt jetzt über mehrere Funktionen, die Ihnen helfen, unstrukturierte Daten zu laden. Zu diesen Funktionen gehören die Folgenden:

- Pivot
- Vereinigung
- Dateninterpreter
- Spalten zusammenführen

Wenn Sie die Datenaufbereitung ohne weitere Vorbereitung ausführen, a) bleiben Sie im Flow Ihrer Analyse und b) können Benutzer ohne Zugriff auf andere Tools die Analyse ihrer Daten in Tableau fortsetzen. Diese Vorgänge sind im Allgemeinen jedoch sehr rechenaufwendig (insbesondere bei großen Datenvolumen) und können die Leistung für Berichtbenutzer beeinträchtigen. Es wird empfohlen, soweit möglich, einen Datenextrakt zu verwenden, um die Daten nach diesen Vorgängen zu materialisieren.

Sie sollten diese Funktionen auch als Ergänzung zu vorhandenen ETL-Prozessen betrachten und überlegen, ob die vorgelagerte Aufbereitung der Daten ein besserer und effizienterer Ansatz ist. Dies gilt insbesondere, falls die Daten in mehreren Berichten oder in anderen BI-Tools verwendet werden.

Datenextrakte

Bisher wurden Techniken zur Verbesserung der Leistung von Datenverbindungen erörtert, bei denen die Daten im ursprünglichen Format bleiben. Sie werden als Direktdatenverbindungen bezeichnet und hinsichtlich Leistung und Funktionen besteht eine Abhängigkeit von der Quelldatenplattform. Zur Verbesserung der Leistung bei Direktverbindungen ist es häufig erforderlich, Änderungen an der Datenquelle vorzunehmen. Bei vielen Kunden ist dies einfach nicht möglich.

Eine für alle Benutzer verfügbare Alternative ist die Nutzung der schnellen Daten-Engine von Tableau, um Daten aus dem Quelldatensystem in einen Tableau-Datenextrakt zu extrahieren. Dies ist für die meisten Benutzer der schnellste und einfachste Weg, die Leistung einer Arbeitsmappe mit einer beliebigen Datenquelle signifikant zu verbessern.

Ein Extrakt ist:

- Ein dauerhafter Datencache, der auf die Festplatte geschrieben wird und reproduzierbar ist
- Ein Datenspeicher in Spalten – ein Format, in dem Daten für analytische Abfragen optimiert wurden
- Während der Abfrage vollständig getrennt von der Datenbank Der Extrakt ist letztendlich ein Ersatz für die Direktdatenverbindung.
- Aktualisierbar, entweder durch erneutes Generieren des Extrakts oder durch inkrementelles Hinzufügen von Datenzeilen zu einem vorhandenen Extrakt
- Architekturorientiert – Anders als die meisten speicherinternen Technologien ist diese nicht durch die Größe des physisch verfügbaren RAM eingeschränkt.
- Übertragbar – Extrakte werden als Dateien gespeichert, können also auf eine lokale Festplatte kopiert und verwendet werden, wenn der Benutzer nicht mit dem Unternehmensnetzwerk verbunden ist. Sie können auch verwendet werden, um Daten in Arbeitsmappenpakete einzubetten, die zur Verwendung mit Tableau Reader weitergegeben werden.
- Häufig viel schneller als die zugrunde liegende Direktdatenverbindung

Tom Brown von The Information Lab hat einen ausgezeichneten Artikel geschrieben, indem er mehrere Anwendungsfälle erläutert, in denen Extrakte von Vorteil sind (lesen Sie unbedingt auch die Kommentare mit zusätzlichen Beispielen von anderen Benutzern):

<http://bit.ly/1F2iDnT>

Ein wichtiger Aspekt der Datenextrakte soll hier betont werden: Sie sind kein Ersatz für ein Data Warehouse, eher als Ergänzung. Sie können zwar verwendet werden, um Daten im Laufe der Zeit zu sammeln und zu aggregieren (d. h. Daten in periodischen Zyklen inkrementell hinzuzufügen). Dies sollte allerdings als taktische, nicht als langfristige Lösung eingesetzt werden. Inkrementelle Aktualisierungen unterstützen keine Aktualisierungs- oder Löschkaktionen an Datensätzen, die bereits verarbeitet wurden. Solche Änderungen erfordern ein erneutes Laden des Extrakts.

Schließlich können Extrakte nicht über OLAP-Datenquellen wie SQL Server Analysis Services oder Oracle Essbase erstellt werden. Die Ausnahme von dieser Regel ist, dass Sie Extrakte von SAP BW erstellen können (siehe diesbezüglichen Abschnitt weiter oben).

[Wann sollte ich Extrakte verwenden? Wann sollte ich Direktverbindungen verwenden?](#)

Extrakte haben wie viele andere Dinge ihren Einsatzzeitpunkt und Einsatzort. Es folgen einige Szenarios, in denen Extrakte von Vorteil sein können:

- Langwierige Abfrageausführung – Wenn Ihr Quelldatensystem die von Tableau Desktop generierten Abfragen nur langsam verarbeitet, kann die Erstellung eines Extrakts eine einfache Möglichkeit sein, die Leistung zu verbessern. Das Datenformat des Extrakts ist speziell für schnelle Reaktion auf analytische Abfragen konzipiert. Sie können sich den Extrakt als einen Cachespeicher zur Abfragebeschleunigung vorstellen. Bei einigen Verbindungstypen ist dies eine empfohlene bewährte Verfahrensweise (zum Beispiel bei großen Textdateien, benutzerdefinierten SQL-Verbindungen), und einige Quellen funktionieren ausschließlich mit diesem Modell (siehe Abschnitt über Cloud-Datenquellen).
- Offline-Analyse – Wenn Sie mit Daten arbeiten müssen, während die ursprüngliche Datenquelle nicht verfügbar ist (zum Beispiel wenn Sie von Ihrem Netzwerk getrennt sind, während Sie reisen oder von zu Hause aus arbeiten). Datenextrakte werden als Datei gespeichert, die einfach auf ein tragbares Gerät wie etwa einen Laptop kopiert werden können. Es ist einfach, zwischen einem Extrakt und einer Direktverbindung hin- und herzuwechseln, falls Sie manchmal eine Netzwerkverbindung haben und manchmal nicht.
- Arbeitsmappenpakete für Tableau Reader/Online/Public – Wenn Sie beabsichtigen, Ihre Arbeitsmappen für andere Benutzer freizugeben, die sie in Tableau wieder öffnen, oder wenn Sie beabsichtigen, sie in Tableau Online oder Tableau Public zu veröffentlichen, müssen Sie die Daten in eine Arbeitsmappenpaketdatei einbetten. Auch wenn die Arbeitsmappe Datenquellen verwendet, die eingebettet werden können (zum Beispiel Dateidatenquellen), sind Datenextrakte für hohe Datenkomprimierung konzipiert, sodass das Arbeitsmappenpaket wesentlich kleiner ist.
- Zusätzliche Funktionen – Es gibt Funktionen in Tableau Desktop, die bei einigen Datenquellen (zum Beispiel Dateidatenquellen über den veralteten MS JET-Treiber) nicht unterstützt werden (zum Beispiel die Vorgänge median/count distinct/rank/percentile aggregations, set IN/OUT usw.). Die Daten zu extrahieren, ist eine einfache Möglichkeit, diese Funktion zu aktivieren.
- Datensicherheit – Wenn Sie eine Teilmenge der Daten von dem Quelldatensystem freigeben möchten, können Sie einen Extrakt erstellen und diesen für andere Benutzer freigeben. Sie können die einzubeziehenden Felder bzw. Spalten einschränken sowie aggregierte Daten freigeben, bei denen Benutzer nur Zusammenfassungswerte sehen sollen, jedoch nicht die individuellen Daten auf Datensatzebene.

Extrakte sind zwar leistungsstark, aber auch nicht die ideale Lösung für alle Probleme. Es gibt einige Szenarien, für die Extrakte möglicherweise nicht geeignet sind:

- Echtzeitdaten – Da Extrakte einen Datenschnappschuss zu einem bestimmten Zeitpunkt darstellen, sind sie nicht geeignet, wenn Sie für Ihre Analyse Echtzeitdaten benötigen. Es ist möglich, Extrakte mithilfe von Tableau Server automatisch zu aktualisieren. Viele Kunden tun dies mehrmals am Tag, aber ein wahrer Echtzeitdatenzugriff erfordert eine Direktverbindung.
- Riesige Datenmenge – Wenn das Datenvolumen, das Sie für Ihre Arbeit benötigen, riesig ist (die Definition von „riesig“ variiert von Benutzer zu Benutzer, aber im Allgemeinen handelt es sich dabei um Millionen bis Milliarden Datensätze), ist das Extrahieren möglicherweise nicht praktisch. Die resultierende Extraktionsdatei kann extrem groß sein oder das Extraktionsverfahren kann sehr viele Stunden in Anspruch nehmen. Beachten Sie, dass es für diese Richtlinie Ausnahmen gibt. Wenn Sie einen riesigen Quelldatensatz haben, jedoch mit einer gefilterten, auf Stichproben basierenden und/oder aggregierten Teilmenge dieser Daten arbeiten, kann die Nutzung eines Extrakts letztendlich eine gute Idee sein. Allgemein ausgedrückt, wurde die Extrakt-Engine von Tableau so konzipiert, dass sie mit einigen 100

Millionen Datensätzen gut funktioniert, aber dies wird von der Form und der Kardinalität Ihrer Daten beeinflusst.

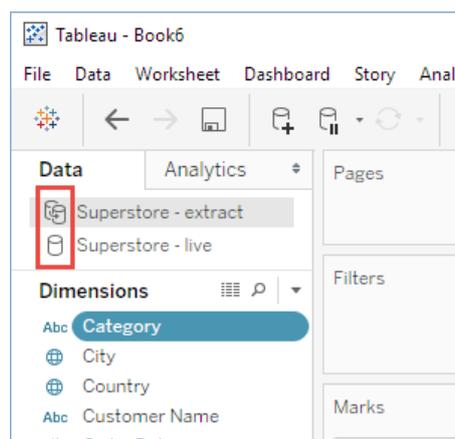
- Pass-Through-RAWSQL-Funktionen – Wenn Ihre Arbeitsmappe Pass-Through-Funktionen verwendet, funktionieren diese nicht mit einem Datenextrakt.
- Robuste Sicherheit auf der Benutzerebene – Wenn Sie eine robust verstärkte Sicherheit auf Benutzerebene benötigen, muss dies in der Datenquelle implementiert werden. Wenn Sie auf der Benutzerebene Filter auf ihre Arbeitsmappe anwenden, können diese jederzeit von einem Benutzer entfernt werden, sodass alle Daten im Extrakt zugänglich werden. Eine Ausnahme dieser Richtlinie bildet ein Extrakt, der mit definierten Datenquellenfiltern auf Tableau Server veröffentlicht wurde, wenn andere Benutzer über den Data Server auf diesen Extrakt zugreifen. Hinweis: Sie müssen sicherstellen, dass die Download-Berechtigungen der Benutzer widerrufen werden, damit sichergestellt ist, dass sie die erzwungenen Filter nicht umgehen können.

Extrakte in Tableau Desktop erstellen

In den meisten Fällen wird ein Extrakt zunächst in Tableau Desktop erstellt, was ein sehr einfacher Vorgang ist. Nachdem Sie eine Verbindung zu Ihren Daten hergestellt haben, wechseln Sie zum Menü „Daten“ und klicken auf „Daten extrahieren“ – anschließend akzeptieren Sie die Standardwerte im Dialogfeld (Details dazu weiter unten). Tableau fragt Sie, ob Sie diesen Extrakt speichern möchten. Wählen Sie einen beliebigen Speicherort für die Datei, obwohl Tableau Ihnen „Eigenes Tableau-Repository | Datenquellen“ vorschlagen wird, was Sie natürlich akzeptieren können!

Warten Sie nun, bis der Extrakt erstellt ist. Wie lange Sie warten müssen, ist abhängig von der verwendeten Datenbanktechnik, der Netzwerkgeschwindigkeit, den Datenvolumen usw. Es ist auch abhängig von der Geschwindigkeit und der Ausstattung Ihrer Workstation, da das Erstellen eines Extrakts speicher- und prozessorintensiv ist.

Den Abschluss des Vorgangs erkennen Sie daran, dass sich das Datenquellsymbol ändert: Es erscheint ein anderes Datenbanksymbol dahinter. Es kennzeichnet eine „Kopie“, was ja auf einen Extrakt genau zutrifft.



Wenn Sie einen Extrakt auf diese Weise erstellen (über Tableau Desktop), erfolgt die Verarbeitung auf Ihrer Workstation. Sie müssen also sicherstellen, dass ausreichende Kapazitäten für diese Aufgabe vorhanden sind. Die Extrakterstellung verwendet alle Ressourcentypen: CPU, RAM, Plattenspeicher, Netzwerk-E/A. Die Verarbeitung großer Datenvolumen auf einem kleinen PC kann Fehler verursachen, falls eines der Elemente nicht ausreichend vorhanden ist. Es wird empfohlen, große Extrakte auf einer geeigneten Workstation auszuführen: schnelle Multicore-CPU, viel RAM, schnelle E/A usw.

Die Erstellung eines Extrakts erfordert temporären Plattenspeicher, um Arbeitsdateien zu schreiben: etwa die doppelte Größe der endgültigen Extraktdatei. Dieser Arbeitsspeicherplatz befindet sich mithilfe der Umgebungsvariablen TEMP angegebenen Verzeichnis (normalerweise C:\WINDOWS\TEMP oder C:\Users\USERNAME\AppData\Local\Temp). Wenn dieses Laufwerk nicht über ausreichend Platz verfügt, legen Sie mithilfe der Umgebungsvariablen einen größeren Speicherort fest.

Falls es nicht möglich sein sollte (oder einfach unpraktisch ist), ein Erstextraktverfahren auf einer Workstation auszuführen, kann mit der folgenden Behelfslösung ein Leerextrakt erstellt werden, der anschließend auf Tableau Server veröffentlicht wird. Erstellen Sie ein berechnetes Feld, `DateTrunc("minute", now())` das enthält. Fügen Sie es den Extraktfiltern hinzu und schließen Sie den Einzelwert aus, den es anzeigt. Dabei müssen Sie schnell vorgehen, weil dieser Filter nach einer Minute nicht mehr gültig ist. Falls Sie mehr Zeit benötigen, erweitern Sie einfach das Veröffentlichungsintervall (runden Sie gegebenenfalls auf 5 oder 10 Minuten bzw. 1 Stunde auf). Damit wird ein leerer Extrakt auf Ihrem Desktop erstellt. Wenn Sie auf dem Server veröffentlichen und den Aktualisierungsplan auslösen, wird der vollständige Extrakt aufgefüllt, weil der ausgeschlossene Zeitstempel nicht mehr derselbe ist.

Extrakte mithilfe der Datenextrakt-API erstellen

Tableau bietet auch eine Schnittstelle zur Anwendungsprogrammierung (API), damit Entwickler direkt eine Tableau-Datenextraktdatei (TDE) erstellen können. Entwickler können diese API einsetzen, um Extrakte aus Software vor Ort und Software-as-a-Service zu generieren. Mit dieser API können Sie Daten systematisch in Tableau verarbeiten, wenn kein nativer Datenkonnektor zur verwendeten Datenquelle vorhanden ist.

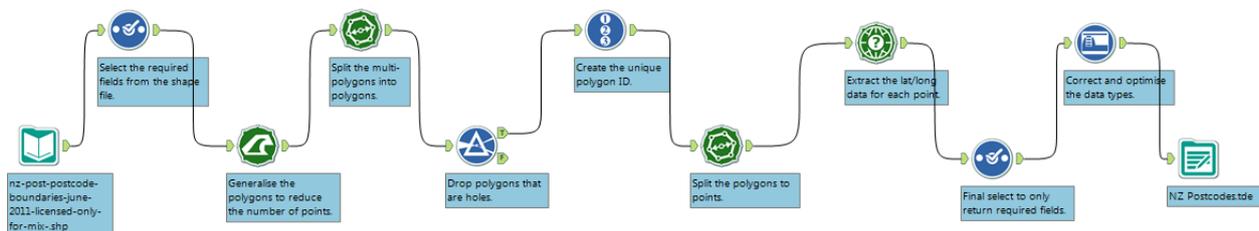
Diese API ist für Entwickler in Python und C/C++/Java sowohl für Windows als auch für Linux verfügbar. Weitere Informationen über die API finden Sie hier:

http://onlinehelp.tableau.com/current/pro/desktop/de-de/extracting_TDE_API.html

Extrakte mit Tools externer Anbieter erstellen

Viele Entwickler bei externen Tool-Anbietern haben die Datenextrakt-API verwendet, um ihren Anwendungen einen native TDE-Ausgabe hinzuzufügen. Zu diesen Anwendungen gehören Analyseplattformen wie Adobe Marketing Cloud sowie ETL-Tools wie Alteryx und Informatica.

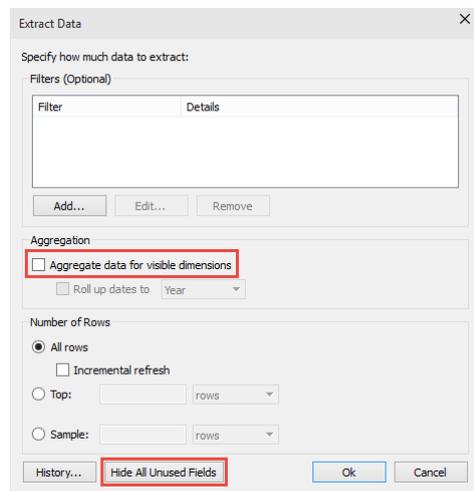
Wenn Sie komplexe Datenaufbereitung benötigen, können Tools wie Alteryx und Informatica verwendet werden, um die ETL-Stufen effizient auszuführen und die aufbereiteten Daten dann direkt in eine TDE-Datei auszugeben, die in Tableau Desktop verwendet werden kann.



Extrakte aggregieren

Die Verwendung eines aggregierten Extrakts kann die Leistung immer verbessern. Selbst wenn Sie Teradata oder Vertica mit riesigen Datenmengen verwenden, kann das Extrahieren von Daten zu einer Verbesserung führen, sofern Sie die Daten in geeigneter Weise aggregieren und filtern. Zum Beispiel können Sie die Daten filtern, wenn Sie nur an den neuesten Daten interessiert sind.

Sie können den Extrakt im Voraus definieren, indem Sie die gewünschten Felder auswählen und das Kontrollkästchen „Daten für sichtbare Dimensionen aggregieren“ im Dialogfeld „Daten extrahieren“ in Tableau Desktop aktivieren. Es besteht auch die Möglichkeit, nach Ausführen Ihrer Analyse und Aufbau Ihres Dashboards, wenn Sie bereit zur Veröffentlichung sind, zum Dialogfeld „Daten extrahieren“ zu wechseln und auf die Schaltfläche „Nicht verwendete Felder ausblenden“ zu klicken. Wenn Sie dann die Daten extrahieren, wird nur das absolute Minimum erforderlich sein, um die Ansicht zu erstellen.



Das erstellen von Aggregatextrakten ist eine sehr leistungsstarke Technik, wenn Sie eine große Menge von Basisdaten haben, jedoch Zusammenfassungen erstellen müssen, die Abfragen im gesamten Datenbestand ausführen. Beispiel: Sie haben Milliarden Datensätze mit detaillierten Transaktionsdaten, die 10 Jahre Umsatz repräsentieren. Sie wollen beginnen, indem Sie den Umsatztrend für alle 10 Jahre anzeigen. Die Abfrage für diese Ausgangsansicht wäre potenziell langsam, weil eine Abfrage in den Milliarden Zeilen ausgeführt werden muss. Wenn Sie einen Extrakt erstellen, der auf Jahresebene aggregiert ist, kann der Abfrageaufwand zum Ansichtszeitpunkt reduziert werden, weil der Extrakt nur 10 Zahlen enthält. Natürlich ist dies ein extrem vereinfachtes Beispiel. Im konkreten Fall hätten Sie mehr Dimensionen als nur die Zeit, aber der Effekt, die Anzahl der zum Ansichtszeitpunkt abzufragenden Datensätze signifikant zu reduzieren, ist erheblich.

Wir können recht ausgefeilte Arbeitsmappen mit mehreren Detailgenauigkeiten erstellen, indem wir mehrere aggregierte Extrakte generieren, die jeweils auf eine bestimmte Detailgenauigkeit abgestimmt sind, oder indem wir aggregierte Extrakte mit Direktverbindungen kombinieren. Angenommen, Sie haben einen Satz anfänglicher Zusammenfassungen, die einen stark aggregierten Extrakt verwenden. Zum Aufrufen von Detaildaten verwenden Sie Aktionsfilter von den Zusammenfassungen zu einem anderen Blatt, das eine Direktverbindung nutzt. Das bedeutet, dass die Zusammenfassungen schnell sind, weil sie nicht den gesamten, zugrunde liegenden Datenbestand zu durchsuchen brauchen. Außerdem brauchen wir nicht die gesamten Basisdaten zu extrahieren, um Drilldown-Aktionen zu unterstützen. Die Direktverbindung ist schnell, weil wir auf der Detailebene nur auf einen kleinen Bestand von Datensätzen zugreifen.

So können Sie auf verschiedenen Ebenen Zahlen kombinieren, anpassen sowie aggregieren, um fast alle Leistungsprobleme zu lösen und so schnell wie möglich zu Ergebnissen zu kommen. Da Tableau den Arbeitsspeicher sehr effizient nutzt, ist die Verbesserung der Performance auf diese Weise in der Regel relativ einfach und Sie können gleichzeitig mehrere Extrakte ausführen.

Extrakte optimieren

Tableau Server optimiert nicht nur die physischen Spalten in der Datenbank, sondern auch die zusätzlichen Spalten, die in Tableau erstellt werden. Diese Spalten umfassen die Ergebnisse von deterministischen Berechnungen, beispielsweise Vorgänge mit Zeichenfolgen und Verkettungen, bei denen sich das Ergebnis nie ändert, sowie Gruppen und Sätze. Die Ergebnisse von nicht-deterministischen Berechnungen, beispielsweise jenen, die einen Parameter oder Aggregationen (wie „Summe“ oder „Mittelwert“) einbeziehen und zur Laufzeit berechnet werden, können nicht gespeichert werden.

Es kann sein, dass ein Benutzer einen Extrakt aktualisiert, nachdem nur zwei Datenzeilen hinzugefügt wurden, und feststellt, dass der Extrakt von 100 MB auf 120 MB gewachsen ist. Dieser Sprung ist auf die Optimierung zurückzuführen. Dabei werden zusätzliche Spalten eingefügt, die berechnete Feldwerte enthalten, weil es wirtschaftlicher ist, Daten auf der Festplatte zu speichern und nicht bei jedem Mal neu zu berechnen, wenn sie benötigt werden.

Worauf Sie achten müssen: Wenn Sie mehrere Kopien einer Verbindung mit einem Datenextrakt herstellen, müssen Sie sicherstellen, dass alle berechneten Felder in der Verbindung enthalten sind, die Sie für die Optionen „Optimieren“ oder „Aktualisieren“ auswählen. Tableau wird sonst Felder nicht materialisieren, von denen es glaubt, dass sie nicht verwendet werden. Es empfiehlt sich, alle berechneten Felder in der primären Datenquelle zu definieren und nach Bedarf in die anderen Verbindungen zu kopieren. Danach aktualisieren oder optimieren Sie den Extrakt nur aus der primären Datenquelle.

Vorbehalt: Eine vertrauenswürdige Ressource in unserem Entwicklungsteam hat Folgendes kommentiert: Es ist zwar möglich, mehrere Verbindungen zu einer einzelnen TDE zu erstellen, dies war jedoch nie ausdrücklich vorgesehen. Daraus können zahlreiche Probleme entstehen, wenn die Verbindungen nicht mehr mit der TDE-Struktur synchron sind. Sein Rat: Verwenden Sie diese Möglichkeit nicht.

Extrakte aktualisieren

Wenn Sie in Tableau Desktop einen Extrakt aktualisieren möchten, wählen Sie Menüoptionen (Menü „Daten“ > [Ihre Datenquelle] > Extrakt > Aktualisieren). Damit werden die Daten aktualisiert und neue Zeilen werden hinzugefügt. In Tableau Server können Sie hingegen während oder nach dem Veröffentlichungsprozess einen Zeitplan anhängen, der von einem Administrator definiert wird, um den Extrakt automatisch zu aktualisieren. Das geringste Inkrement für den Zeitplan beträgt 15 Minuten. Mit dem Zeitplan kann die Aktualisierung täglich, wöchentlich etc. zu einem bestimmten Zeitpunkt erfolgen. Sie können ein „bewegliches Fenster“ einrichten, um die Daten laufend auf die jüngsten Werte zu aktualisieren.

Hinweis: Wenn Sie Ihre Daten öfter als alle 15 Minuten aktualisieren möchten, sollten Sie eine Direktverbindung in Betracht ziehen oder eine synchronisierte Berichtsdatenbank einrichten.

Sie können zwei Aktualisierungszeitpläne für einen einzelnen Extrakt auswählen.

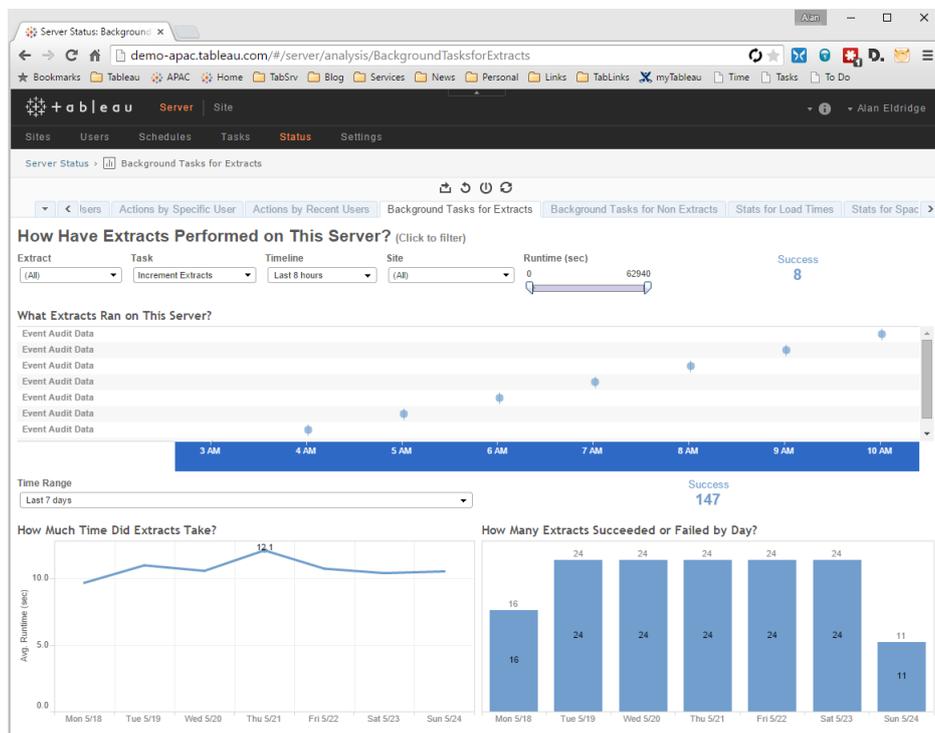
- Bei einer inkrementellen Aktualisierung werden nur Zeilen hinzugefügt. An bestehenden Zeilen werden keine Änderungen vorgenommen.
- Eine vollständige Aktualisierung verwirft den aktuellen Extrakt und generiert einen neuen aus der Datenquelle.

Was geschieht, wenn die Aktualisierung länger dauert als das Aktualisierungsintervall?
 Wenn die Aktualisierung eines Extrakts länger dauert als das Inkrement, werden die dazwischen liegenden Aktualisierungen übersprungen. Beispiel: Der Zeitplan ist darauf eingestellt, die Daten einmal pro Stunde zu aktualisieren, aber die Datenmenge ist so groß, dass die Aktualisierung anderthalb Stunden dauert. In diesem Fall geschieht Folgendes:

- Die erste Aktualisierung beginnt um 1:00 und endet um 2:30.
- Die nächste Aktualisierung sollte um 2:00 Uhr starten. Da die erste jedoch noch läuft, wird sie übersprungen.
- Die nächste Aktualisierung beginnt um 3:00 und endet um 4:30.

Extrakte pflegen

Die Wartungsbildschirme melden, welche Hintergrundaufgaben gerade laufen, und welche in den letzten 12 Stunden gelaufen sind. Die Farbcodierung wird verwendet, um den Status dieser Aufgaben zu zeigen. Die Wartungsbildschirme sind für Administratoren und andere entsprechend berechnete Benutzer verfügbar, die die Berechtigung haben können, eine Ad-hoc-Aktualisierung eines Extrakts vorzunehmen. Wenn beispielsweise eine Datenbank gerade lädt, können Sie einen Trigger setzen, um einen Extrakt einzuleiten, nachdem die Datenbank geladen ist.



Sie können eine Arbeitsmappe inkrementell oder vollständig über das Befehlszeilentool aktualisieren, wenn Sie Tableau Server verwenden, oder die Tableau.exe-Befehlszeile, wenn Sie Tableau Desktop verwenden. Wenn Sie komplexe Anforderungen an die Zeitplanung haben, können Sie sie von einem externen Zeitplanungstool aufrufen, beispielsweise mit „Aufgabenplanung“ unter Windows. Dieser Ansatz ist erforderlich, wenn Sie einen Aktualisierungszyklus haben, der kürzer ist als das Minimum von 15 Minuten, das über die Tableau Server-Schnittstelle verfügbar ist.

Benutzer von Tableau Online können den Synchronisationsclient verwenden, um Datenquellen am Standort über Zeitpläne auf dem neuesten Stand zu halten, die im Tableau Online-Service definiert wurden. Weitere Informationen über dieses Tool finden Sie hier:

http://onlinehelp.tableau.com/current/online/de-de/qs_refresh_local_data.htm

Data Governance

Es ist zwar selbst keine Datenquelle, aber ein anderer Weg, um eine Verbindung zu Datenquellen herzustellen: der Data Server von Tableau Server. Der Data Server unterstützt sowohl Datenverbindungen als auch Datenextrakte und bietet mehrere Vorteile gegenüber eigenständigen Datenverbindungen:

- Da die Metadaten zentral auf dem Tableau Server gespeichert werden, können sie von mehreren Arbeitsmappen und mehreren Autoren/Analysten gemeinsam genutzt werden. Die Arbeitsmappen enthalten einen Verweis auf die zentralisierte Metadatendefinition und jedes Mal, wenn sie geöffnet werden, prüfen sie, ob Änderungen vorgenommen wurden. Ist dies der Fall, wird der Benutzer aufgefordert, die in die Arbeitsmappe eingebettete Kopie zu aktualisieren. Das bedeutet, dass Änderungen an der Geschäftslogik nur an einem Ort durchgeführt werden müssen und anschließend an alle abhängigen Arbeitsmappen propagiert werden können.
- Wenn es sich bei der Datenquelle um einen Datenextrakt handelt, kann dieser in verschiedenen Arbeitsmappen verwendet werden. Ohne den Data Server würde jede Arbeitsmappe ihre eigene lokale Kopie des Extrakts enthalten. Dadurch wird die Anzahl der redundanten Kopien reduziert, was wiederum den auf dem Server erforderlichen Speicherplatz mindert und mehrfache Aktualisierungsprozesse vermeidet.
- Wenn es sich bei der Datenquelle um eine Direktverbindung handelt, brauchen die Treiber für die Datenquelle nicht auf jedem Analysten-PC installiert zu werden, sondern nur auf dem Tableau Server. Der Data Server agiert für Abfragen von Tableau Desktop wie ein Proxy.

Liegt es an meiner Umgebung?

Es gibt Fälle, in denen eine Arbeitsmappe beim Einzelbenutzertest zufriedenstellend abläuft, auf Tableau Server (oder Tableau Online) jedoch – mit vielen Benutzern – langsam ausgeführt wird. Im folgenden Abschnitt werden Bereiche aufgezeigt, in denen die Unterschiede zwischen Einzel- und Mehrbenutzerszenarien unterschiedliche Leistungsniveaus aufweisen.

Upgrade durchführen

Das Entwicklungsteam bei Tableau befasst sich ständig mit der Verbesserung der Leistung und Benutzerfreundlichkeit unserer Software. Ein Upgrade auf die neueste Version von Tableau Server kann gelegentlich signifikante Verbesserungen bei Leistung und Stabilität zur Folge haben, ohne Änderungen an der Arbeitsmappe vornehmen zu müssen.

Sehen Sie sich die Versionshinweise von Tableau an und aktualisieren Sie auf die neueste Version:

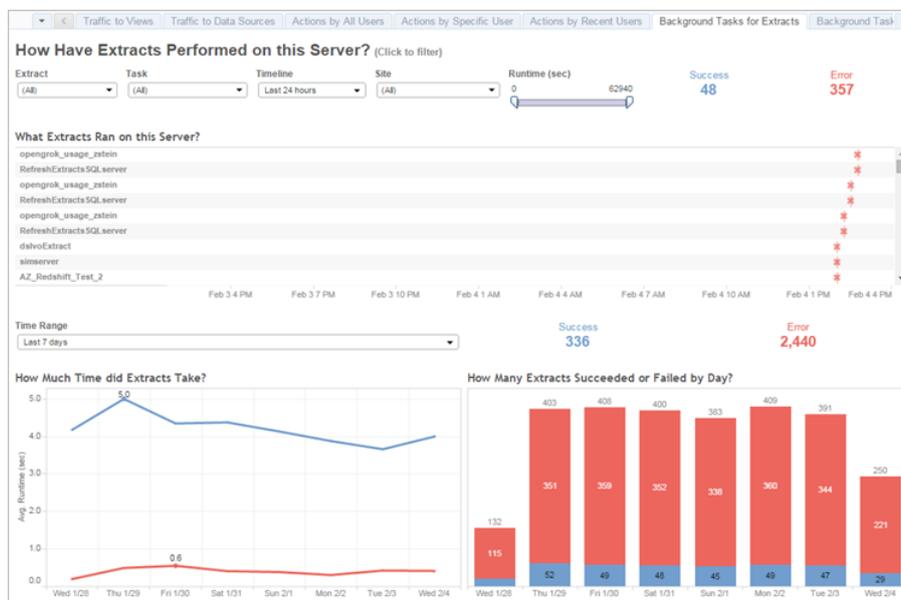
<http://www.tableau.com/de-de/support/releases>

Testen Sie Tableau Desktop auf dem Server

Es treten Situationen auf, in denen eine Arbeitsmappe in Tableau Desktop auf Ihrer Workstation eine gute Leistung aufweist, jedoch über Tableau Server unzureichend ausgeführt wird. Wenn Sie die Arbeitsmappe in einer Kopie von Tableau Desktop öffnen, das auf dem Tableau Server-Rechner installiert ist, können Sie ermitteln, ob es an Ihrer Arbeitsmappe oder an der Konfiguration mit dem Server liegt. Bei diesem Ansatz lässt sich ermitteln, ob ein Treiber inkompatibel ist, oder ob ein Problem mit dem Netzwerk besteht, etwa schlechte Konfiguration bei Routing, DNS oder Proxy.

Separate Aktualisierungen und interaktive Arbeitslasten

Sollte die Serverleistung langsam sein, entnehmen Sie der Verwaltungsansicht „Hintergrundaufgaben“ Ihre aktuellen Zeitpläne für Aktualisierungsaufgaben.



Planen Sie Aktualisierungen nach Möglichkeit für Nebenzeiten. Wenn Ihre Hardwarekonfiguration dies zulässt, können Sie die Hintergrundprozesskomponenten auf einen spezifischen Worker-Knoten verschieben.

Tableau Server überwachen und feinabstimmen

Tableau Server umfasst einige Ansichten für Administratoren, die die Überwachung der Tableau Server-Aktivitäten erleichtern sollen. Diese Ansichten befinden sich in der Tabelle „Analyse“ auf der Wartungsseite des Servers.

Views	Analysis
Traffic to Views	View count, viewers, and viewer behavior for published views.
Traffic to Data Sources	Data source usage, users, and user behavior for published data sources.
Actions by All Users	Actions for all users.
Actions by Specific User	Actions for a specific user, including items used.
Actions by Recent Users	Recent actions by users, including last action time and idle time.
Background Tasks for Extracts	Completed and pending extract task details.
Background Tasks for Non Extracts	Completed and pending background task details (non-extract).
Stats for Load Times	View load times and performance history.
Stats for Space Usage	Space used by published workbooks and data sources, including extracts and live connections.

Weitere Informationen zu diesen Ansichten finden Sie unter dem folgenden Link:

<http://onlinehelp.tableau.com/current/server/de-de/adminview.htm>

Zusätzliche benutzerdefinierte Verwaltungsansichten können Sie erstellen, indem Sie eine Verbindung zur Datenbank PostgreSQL herstellen, einem Teil des Tableau-Repositorys. Anleitungen finden sich hier:

http://onlinehelp.tableau.com/current/server/de-de/adminview_postgres.htm

Überprüfen Sie das Zeitüberschreitungslimit der VizQL-Sitzung

Standardmäßig ist das Zeitüberschreitungslimit der VizQL-Sitzung auf 30 Minuten eingestellt. Selbst VizQL-Sitzungen, die sich im Leerlauf befinden, verbrauchen weiterhin Speicher und CPU-Zyklen.

Wenn ein niedrigeres Limit ausreichend ist, ändern Sie mit tabadmin die Einstellung vizqlserver.session.expiry.timeout:

http://onlinehelp.tableau.com/current/server/de-de/reconfig_tabadmin.htm#ida6864815-db67-4a51-b8b6-c93617582091

Beurteilen Sie die Prozesskonfiguration

Tableau Server ist in viele verschiedene Komponenten unterteilt, die als „Services“ bezeichnet werden. Obwohl deren Standardkonfiguration für eine Vielzahl von Szenarien konzipiert ist, können die Prozesse neu konfiguriert werden, um andere Leistungsziele zu erreichen. Insbesondere können Sie steuern, auf welchen Computern wie viele Prozesse ausgeführt werden. Leitfäden zur Bereitstellung auf einem, zwei und drei Computern finden Sie unter „Beispiele für die Leistungsoptimierung“:

http://onlinehelp.tableau.com/current/server/de-de/perf_extracts_view.htm#idd21e8541-07c4-420f-913e-92dcaf5f0c34

Infrastruktur

64-Bit

Versionen von Tableau Server vor Tableau 10 können auf 32-Bit-Microsoft-Betriebssystemen ausgeführt werden. Den Kunden wird jedoch dringend empfohlen, Ihre Produktionsumgebungen mit der 64-Bit-Version zu installieren. 32-Bit-Installationen werden nur für DEV/TEST-Installationen empfohlen.

Ab Tableau 10 ist Tableau Server nur als 64-Bit-Anwendung verfügbar.

Mehr CPU/RAM hinzufügen

Gleichgültig ob Tableau Server nun auf einem oder mehreren Computern ausgeführt wird: Als allgemeine Regel gilt, dass sich durch mehr CPU-Kerne und mehr RAM die Leistung verbessern lässt. Stellen Sie sicher, dass Sie die empfohlenen Hardware- und Softwareanforderungen von Tableau Server erfüllen (<http://onlinehelp.tableau.com/current/server/de-de/requ.htm#ida4d2bd02-00a8-49fc-9570-bd41140d7b74>), und beurteilen Sie anhand von „Zeitpunkt des Hinzufügens und Neukonfigurierens von Arbeitscomputern“ (http://onlinehelp.tableau.com/current/server/de-de/distrib_when.htm#idfdd60003-298e-47e6-8133-3d2490e21e07), ob weitere Computer hinzugefügt werden sollten.

TabMon (siehe weiter oben in diesem Dokument) ist ein hervorragendes Tool zum Sammeln von Nutzungsdaten, das Ihnen bei der Kapazitätsplanung helfen kann.

Achten Sie auf den E/A-Verkehr

Einige Aktionen von Tableau bewirken intensiven E/A-Verkehr (z. B. Laden/Erstellen/Aktualisieren eines Datenextrakts) und arbeiten optimiert, wenn anstelle von mechanischen Festplatten SSDs verwendet werden. Russell Christopher hat in seinem Blog „Tableau Love“ eine Reihe ausgezeichneter Beiträge veröffentlicht, in denen die Auswirkung von Kernanzahl, CPU-Geschwindigkeit und IOPS auf die allgemeine Leistung untersucht wird. Sein Experiment wurde zwar auf AWS durchgeführt, lässt sich jedoch dennoch auf alle Umgebungen übertragen:

<http://bit.ly/1f17oa3>

<http://bit.ly/1f17n50>

Physisch vs. virtuell

Zahlreiche Kunden stellen jetzt Tableau Server auf virtualisierten Infrastrukturen bereit. Die Virtualisierung erfordert immer einen zusätzlichen Aufwand, ist also nicht so schnell wie eine Installation auf Hardware, obwohl moderne Hypervisor-Technologie diesen zusätzlichen Aufwand signifikant reduziert hat.

Beim Installieren auf virtuellen Maschinen muss unbedingt sichergestellt werden, dass Tableau Server über eigene RAM- und CPU-Ressourcen verfügt. Wenn er mit anderen virtuellen Maschinen im Wettbewerb um Ressourcen auf dem physischen Host steht, kann die Leistung stark beeinträchtigt werden.

Eine gute Ressource für die Feinabstimmung virtueller Bereitstellungen ist das Whitepaper „Deploying Extremely Latency-Sensitive Applications in vSphere 5.5“ von VMWare. Auf Seite 15 gibt es eine Liste der bewährten Verfahrensweisen für Latenz-empfindliche Anwendungen:

<http://vmw.re/1L4Fyr1>

Internet-Browser

Tableau verwendet ausgiebig JavaScript. Die Geschwindigkeit des JavaScript-Interpreters im Browser hat Auswirkungen auf die Schnelligkeit des Renderns. Moderne Browser sind ebenbürtig, aber entwickeln sich schnell weiter. Es lohnt sich jedoch zu prüfen, ob Versionen und Leistung Ihres Browsers Ihre Nutzererfahrung beeinträchtigen kann.

Fazit

Ein Weiser sagte einst (ein wichtiger Rat für mich zu diesem Dokument): „Sag ihnen, was du ihnen sagen wirst. Sag es ihnen. Sag ihnen anschließend, was du ihnen gesagt hast.“ Wirklich ein weiser Rat.

Hier also noch mal die wichtigsten Punkte, die Sie aus der Lektüre dieses Dokuments mitnehmen:

- Es gibt keinen Königsweg. Die Leistung kann aus einer Vielzahl von Gründen schwach sein. Sammeln Sie Daten, um zu ermitteln, wo die Zeit versickert. Konzentrieren Sie sich dann auf Verbesserungen in den teuersten Bereichen, dann im nächsten Bereich usw. Hören Sie auf, wenn es schnell genug ist oder das Kosten-/Nutzenverhältnis nicht mehr vertretbar ist.
- Viele Dashboards werden durch schlechtes Design langsam. Keep it simple. Versuchen Sie nicht, zu viel auf einmal zu zeigen, und verwenden Sie geführte Analysekonzepte, sofern möglich.
- Arbeiten Sie nicht mit Daten, die Sie nicht benötigen. Verwenden Sie Filter, blenden Sie nicht verwendete Felder aus und aggregieren sie.
- Kämpfen Sie nicht gegen die Daten-Engine. Sie ist intelligent und versucht, Ihnen zu helfen. Verlassen Sie sich darauf, dass sie effiziente Abfragen generiert.
- Je sauberer Ihre Daten sind und je besser sie mit der Struktur Ihrer Fragen übereinstimmen, desto schneller laufen Ihre Arbeitsmappen und desto glücklicher wird Ihr Leben sein.
- Extrakte sind eine schnelle und einfache Möglichkeit, um die meisten Arbeitsmappen zu beschleunigen. Wenn Sie keine Echtzeitdaten benötigen und mit Milliarden von Datenzeilen arbeiten, sollten Sie es damit versuchen.
- Zeichenfolgen und Datumsangaben sind langsam, Zahlen und boolesche Werte sind schnell.
- Dieses Dokument basiert zwar auf den besten Ratschlägen von zahlreichen intelligenten Menschen, dennoch sind Empfehlungen eben nichts anderes als Empfehlungen. Sie müssen testen, welche die Leistung in Ihrem speziellen Fall verbessern.
- Wenn Sie mit kleinen Datenvolumen arbeiten, sind die meisten von geringer Bedeutung - Sie können schlechte Leistung mit Hardware totschiessen. Es schadet jedoch nicht, diese Empfehlungen in allen Ihren Arbeitsmappen zu berücksichtigen, weil Sie nie wissen, wann Ihre Daten anwachsen werden.
- Übung macht den Meister.

Jetzt sind Sie an der Reihe, viele effiziente Arbeitsmappen zu entwickeln!