

Katharina Koch

SEO-Data-Warehousing (Teil 1 – für kleinere Projekte)

Als SEO arbeitet man viel mit Daten aus den unterschiedlichsten Quellen. Will man die Performance seiner eigenen Domain analysieren, muss man sich durch unzählige Tools klicken. Das kostet jeden Tag viel Zeit – Zeit, die fehlt, um neue Ideen für das Portal zu entwickeln und umzusetzen.

Wie kann man Zeit bei der Analyse sparen und auch noch die wichtigsten Daten seiner Domain dauerhaft sichern?

Die Lösung: Man baut sich sein eigenes SEO-Data-Warehouse. Katharina Koch zeigt im ersten Teil beispielhaft, wie man sich mit einfachen Bordmitteln bzw. den kostenlosen Google-Tabellen Schritt für Schritt für kleinere eigene Projekte so etwas aufbauen kann. Ein Umsetzungsbeispiel für größere Projekte mit größerem Datenvolumen finden Sie dann im zweiten Teil in der nächsten Ausgabe der Website Boosting.

Wikipedia definiert ein Data-Warehouse als eine „für Analysezwecke optimierte zentrale Datenbank, die Daten aus mehreren [...] Quellen zusammenführt“. Damit stehen die beiden wichtigsten Anforderungen bereits fest: Daten sammeln und für Analysen zur Verfügung stellen.

Womit kann ein Data-Warehouse gebaut werden

Je nach Projekt und Ressourcen kann so eine Datensammlung ganz unterschiedlich aussehen. Für kleine Projekte mit wenig bis keinem Budget sind Google Sheets sicher die richtige Wahl. Es gibt verschiedene Anwendungen, die die Erstellung eines Data-Warehouse erleichtern. Zudem können mehrere Personen gleichzeitig an einer

Google-Tabelle arbeiten. Eine kleine Einschränkung gibt es jedoch zu beachten: Pro Datei können lediglich zwei Millionen Zellen gespeichert werden.

Die SEO-Tools für Excel haben in etwa den gleichen Funktionsumfang und ähnliche Limitierungen wie Google-Tabellen. Zwar gibt es hier kein Zell-Limit, aber es können auch nur maximal eine Million Zeilen pro Tabellenblatt angelegt werden. Zudem kann es zu Problemen mit der Rechenleistung der CPU kommen, sodass man gerade bei großen Excel-Tabellen sehr lange warten muss, bis die Datei geladen wird oder Formeln ausgeführt werden. An einer Excel-Tabelle kann auch immer nur eine Person arbeiten.

Foto: solarseven / thinkstockphotos.de

DIE AUTORIN



Katharina Koch ist Head of SEO für die Auto1 Group GmbH. Datengetriebenes SEO steht in Ihrer Arbeit an erster Stelle. Im Laufe der Jahre hat sie unzählige Reportings erstellt und sich immer neue Wege der Automatisierung ausgedacht.

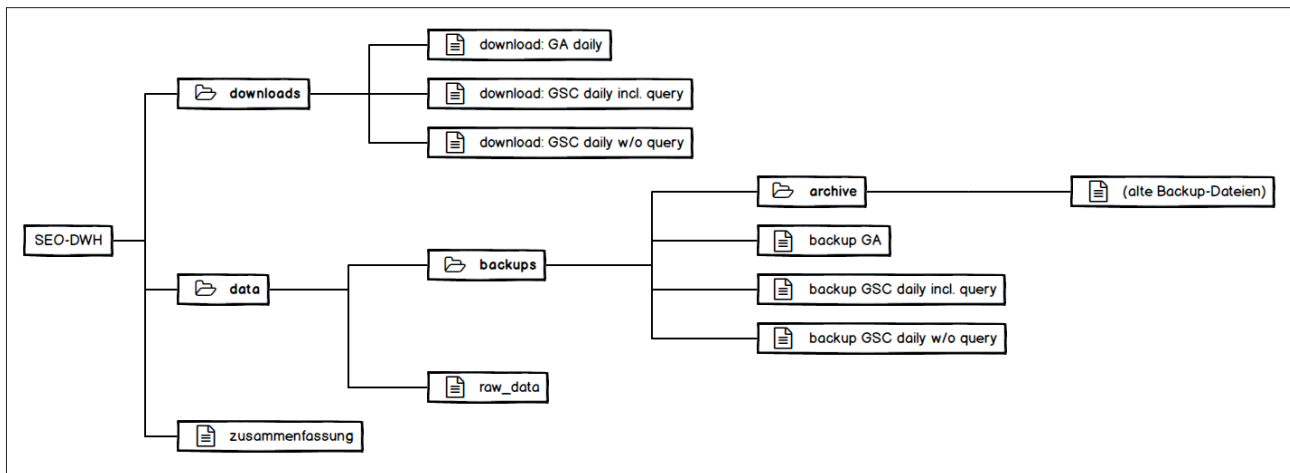


Abb.1: SEO-DWH-Struktur

Datenbanken sind gerade für große bis sehr große Projekte und Datenmengen gut geeignet. Jedoch ist hier technisches Know-how gefragt, um individuelle API-Lösungen mit der Datenbank zu verbinden.

In diesem Artikel wird ein Einblick gegeben, wie man mittels Google-Tabellen ein SEO-Data-Warehouse erstellen kann. Hier sollen in erster Linie Funktionsweisen anhand von Google-Analytics- und Google-Search-Console-Daten aufgezeigt werden. Ziel ist es, die Daten, die durch beide Tools bereitgestellt werden, dauerhaft zu speichern und für eine tägliche Analyse aufzubereiten. Die geschilderte Herangehensweise dient als Beispiel und kann individualisiert oder auf mehr Datenquellen erweitert werden.

Was im SEO-DWH gespeichert wird

Wichtige Indikatoren für die Performance einer URL sind die Daten, die Google seinen Nutzern kostenlos zur Verfügung stellt. Das sind unter anderem aus der Google Search Console die Ranking-Keywords inklusive ihrer Positionen sowie die Klick-Rate. Aus Analytics sind userbasierte KPIs wie Anzahl der Sitzungen auf einer URL, die Rate der neuen Nutzer, Absprungrate, Seiten pro Sitzung und die durchschnittliche Sitzungsdauer interessant.

Dank der von Google und anderen entwickelten Add-ons für Google-Tabellen ist es sehr einfach, die benötigten Daten automatisiert herunterzuladen.

Wenn man anfängt, sich ein SEO-Data-Warehouse zu bauen, sollten mindestens die Datensätze der letzten 90 Tage heruntergeladen werden. Somit kann man gleich mit der Analyse starten und muss nicht warten, bis genügend Daten vorhanden sind. Danach werden die Daten automatisiert täglich für den letzten Tag geladen.

Ein SEO-DWH mit Google Sheets

Bevor man anfängt, eine Datenbank mit Auswertungs-Funktion zu bauen, sollte man sich eine Struktur überlegen. Die Struktur für dieses Data-Warehouse findet sich in Abbildung 1.

Zur besseren Übersicht werden zunächst zwei Unterordner erstellt. Im Ordner „Downloads“ werden die Daten aus Search Console und Analytics heruntergeladen. Im Ordner „data“ werden die heruntergeladenen Daten im „backups“-Ordner abgelegt.

Es ist empfehlenswert, mindestens einmal pro Monat diese Back-up-Dateien im Archiv abzuspeichern und nur aktuell benötigte Datensätze in den Back-up-Files zu belassen, zum Beispiel die letzten 40 Tage. Wie eingangs erwähnt, können Google-Tabellen nur zwei Millionen Zellen speichern, und je größer diese

Dateien werden, umso langsamer ist das Reporting am Ende auch.

Die Back-up-Dateien werden dann in der Tabelle „raw_data“ zusammengeführt und aufbereitet. Von dort aus wird schließlich in der „Zusammenfassung“ ein Reporting auf Tages- und Wochenbasis erstellt.

Für eine gute Analyse werden in diesem Beispiel die Tagesdaten der letzten acht Tage und die Wochendaten der letzten fünf Wochen angezeigt. Um genügend Daten für dieses Reporting parat zu haben, sollten sich dort mindestens die Datensätze der letzten 35 Tage befinden.

Schritt 1: Daten herunterladen

Bevor die Daten aus Google Analytics und Google Search Console heruntergeladen werden können, müssen die entsprechenden Add-ons installiert werden.

Öffnet man eine Google-Tabelle, findet man im Menü den Punkt „Add-ons“. Dort lassen sich unter „Add-ons aufrufen“ neue Tools zu den Google Sheets hinzufügen. Für das SEO-Data-Warehouse sollten die Add-ons „Google Analytics“ für Google Analytics und „Search Analytics for Sheets“ für die Search Console integriert werden.

Beide Add-ons fragen zunächst nach einer Zugriffsberechtigung zum Google-Konto. Hier sollte dasjenige gewählt werden, über welches auf Analytics und Search Console zugegriffen wird.

Schritt 1.1: Google-Analytics-Daten herunterladen

Sind beide Add-ons installiert, kann man damit anfangen, sein DWH zusammenzustellen. In diesem Beispiel wird zunächst eine Datei mit dem Namen „download: GA daily“ im Ordner „downloads“ angelegt.

Unter dem Tabellen-Menü „Add-ons“ finden sich jetzt zwei neue Einträge: „Google Analytics“ und „Search Analytics for Sheets“. Geht man mit dem Mauszeiger auf „Google Analytics“, öffnet sich ein Untermenü. Dort wählt man „Create new report“ aus. Nach dem Klick wird ein neuer Tab „Report Configuration“ erstellt.

In diesem Tab werden alle Reportings angelegt. Dabei entspricht jede Spalte einem Report. Für dieses SEO-DWH wird jedoch nur ein Report benötigt, der immer die Daten für den letzten Tag herunterlädt.

Zur Analyse der URL-Performance in diesem Beispiel werden folgende Metriken heruntergeladen: ga:users, ga:newUsers, ga:sessions, ga:bounces, ga:sessionDuration und ga:pageviews. Als Dimensionen werden ga:landingPagePath, ga:deviceCategory und ga:date ausgewählt.

Unter ga:landingPagePath werden dann die URLs aufgelistet; ga:deviceCategory gibt an, ob die URL über Desktop, Mobile oder Tablet angezeigt wurde; ga:date zeigt das Sitzungsdatum an. Ob man sich das Device anzeigen lassen möchte oder nicht, ist jedem selbst überlassen. Da mobiles Surfen jedoch immer weiter zunimmt, ist es ratsam, sich die Daten am Ende auch auf Device-Level anschauen zu können.

Unter „Filters“ gibt es noch die Möglichkeit, die heruntergeladenen Daten zu spezifizieren. So kann man über „ga:landingPagePath!=(not set)“ alle Daten herausfiltern, die Google als (not set) anzeigt. Sollte man auch Paid-Traffic über seine Landingpages laufen lassen, und möchte man trotzdem nur den orga-

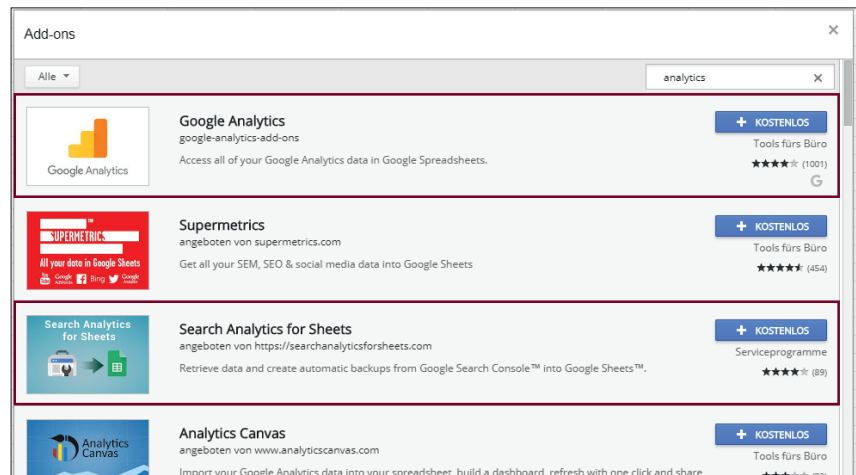


Abb.2: Google Add-ons: Was muss installiert werden?

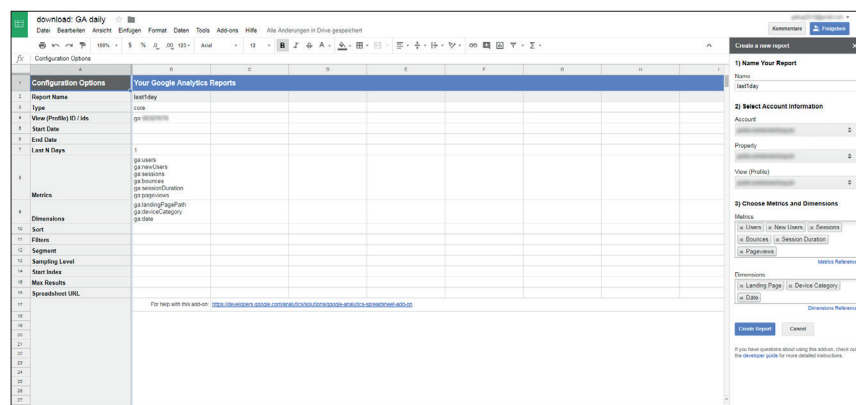


Abb.3: Report Configuration für Google Analytics

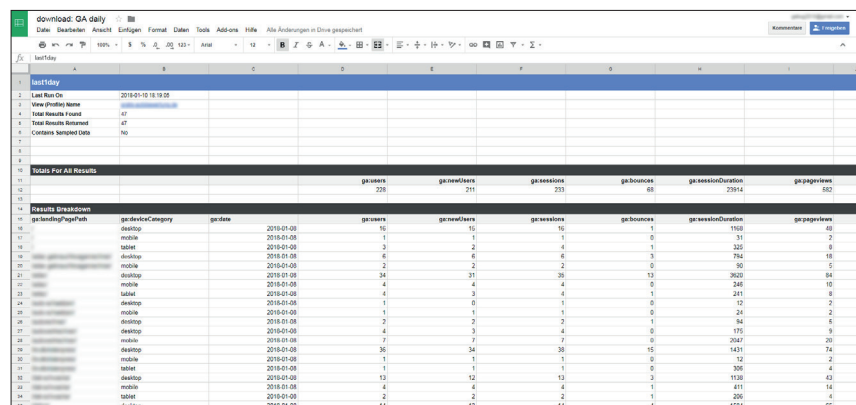


Abb. 4: Google Analytics: heruntergeladener Report in Tab „lastDay“

nischen Traffic analysieren, kann man über „ga:medium==organic“ nur den Teil der User-Daten, die über organische Suchen gekommen sind, herausfiltern.

Da dieses Reporting einmal pro Tag Daten aus Google Analytics herunterladen soll, reicht es, bei „Last N Days“ nur einen Tag auszuwählen. Natürlich ist hier auch jeder andere Zeitraum möglich.

Im Menüpunkt „Add-ons > Google Analytics > Schedule reports“ hat man die Möglichkeit, das erstellte Reporting automatisch herunterladen zu lassen. So kann man das Reporting jede Stunde, jeden Tag, jede Woche oder jeden Monat laufen lassen. Als Standard-Einstellung ist „every day between 3 a.m. - 4 a.m.“ eingestellt.

Wird das Reporting ausgeführt, wird automatisch einmalig ein neuer Tab mit dem Report-Namen erstellt, in diesem Fall „last1day“. Diese Daten werden jedes Mal, wenn das Reporting ausgeführt wird, neu überschrieben. Will man also eine lückenlose Dokumentation seines Traffics haben, müssen diese Daten einmal am Tag aus diesem Sheet kopiert und in einer Back-up-Datei gespeichert werden. Dafür können „Google Script Apps“ eingesetzt werden. Wie das funktioniert, wird in Schritt 2 „Daten sichern“ erklärt.

Schritt 1.2: Google-Search-Console-Daten herunterladen

Für die Daten der Google Search Console ist es ratsam, zwei Download-Dateien anzulegen, eine auf URL-Basis und eine auf Keyword-Basis.

Der Grund dafür ist die Search Console selbst. Im Interface werden maximal 1.000 Zeilen pro Report angezeigt, egal, ob Suchanfragen oder Seiten. Die API ermöglicht es, bis zu 5.000 Zeilen pro Report herunterzuladen – zumindest mit Queries. Ohne Queries können unbegrenzt Zeilen heruntergeladen werden.

Das Add-on „Search Analytics for Sheets“ funktioniert im Hintergrund über die API, damit können auch hier maximal 5.000 Zeilen Daten mit Queries heruntergeladen werden. Bei großen Seiten mit vielen bis sehr vielen Keywords werden so nie alle Keywords heruntergeladen. Allerdings gehen dann nur Keywords mit zehn und weniger Impressions verloren. Wenn man trotzdem so viele Daten wie möglich sammeln möchte, kann man sich auch ein Reporting pro Device erstellen.

Erstellt werden die Reportings über „Add-ons > Search Analytics for Sheets > Open Sidebar“. Unter „Backups“ wird dann die gewünschte Domain ausgewählt, wann das Back-up erstellt werden soll, der Suchtyp und

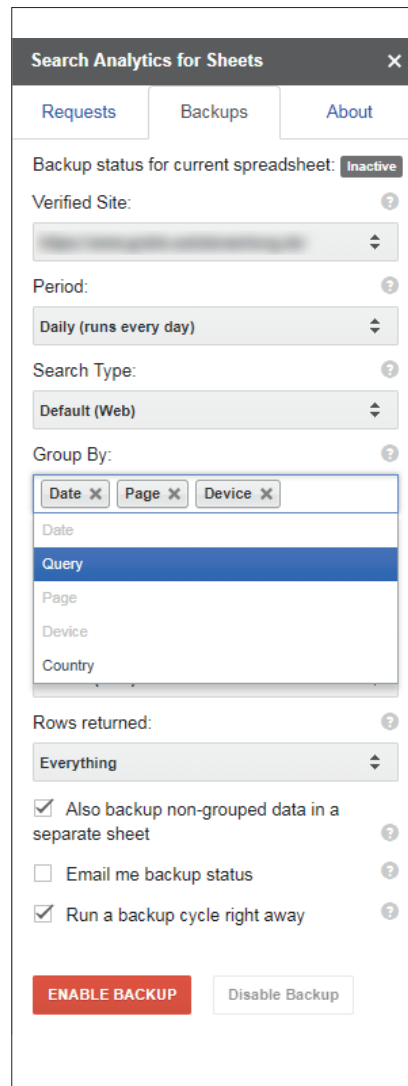


Abb.5: Download Google-Search-Console-Daten

wie die Daten gruppiert werden sollen. Auf Wunsch können auch Filter gesetzt werden.

Leider sind die Back-up-Zeiträume weniger flexibel als die Reportings für Google Analytics. So kann man sich nur zwischen einem täglichen Back-up oder einem Back-up an jedem dritten Tag eines Monats entscheiden. Für dieses Szenario wird ein tägliches Back-up empfohlen. Es ist auch immer nur ein automatischer Download pro Datei möglich.

Als Search Type ist automatisch „Default (Web)“ eingestellt. Unter „Group By“ werden die Dimensionen dieses Reportings festgelegt. Es müs-

sen die gleichen Dimensionen wie zuvor bei Analytics gewählt werden, damit die Datensätze korrekt aufeinander verweisen. Das heißt, in der Datei „downloads: GSC daily w/o query“ werden „Date“, „Page“ und „Device“ ausgewählt, in der Datei „downloads: GSC daily incl. query“ zusätzlich noch „Query“.

Als Letztes kann man sich noch aussuchen, ob der erste Eintrag sofort erstellt werden soll. Hat man den letzten Tag an Anfang bereits heruntergeladen, ist das natürlich nicht notwendig. Mit einem Klick auf „Enable Backup“ wird der Back-up-Prozess gestartet. Das Add-on erstellt hier jeden Tag einen neuen Tab mit dem Namen des Tages, zu dem die Daten im Tab gehören. Zum Beispiel werden alle Daten für den 5. Januar 2018 im Tab 2018-01-05 gespeichert.

Schritt 2: Daten dauerhaft speichern

Hat man seine Daten heruntergeladen, gilt es, sie auch zu speichern, um sie in Zukunft für Analysen parat zu haben. Für dieses Beispiel-DWH wurde dafür ein extra Unterordner im „data“-Ordner erstellt. Der „backups“-Ordner enthält sowohl drei Back-up-Dateien, in denen die Daten aus den automatischen Downloads gespeichert werden, als auch einen „Archive“-Ordner.

Um die Daten aus den täglichen Downloads nicht jeden Tag per Hand von A nach B kopieren zu müssen, wird hier ein „Google Apps Script“ verwendet. Diese bieten dem Anwender die Möglichkeit, Automatisierungs-Skripte selbst zu schreiben.

Da die „Google Apps Scripts“ auf JavaScript basieren, sind erste Kenntnisse in dieser Sprache von Vorteil. Das gesamte hier verwendete Skript inkl. Kommentierungen kann unter: <http://einfach.st/dwskript> als einfache txt-Datei heruntergeladen werden.

Was macht das Skript?

1. Öffnet die Google-Tabelle mit der angegebenen ID (zu finden in der URL der Tabelle zwischen /d/ und /edit#).

```
var source_gsc_with = SpreadsheetApp.openById(,Tabellen-ID');
```

2. Öffnet entweder den aktiven Tab oder den Tab mit dem Namen XY.

```
var sheet_gsc_with = source_gsc_with.getActiveSheet();
//oder
var sheet_ga = source_ga.getSheetByName(,Tab-Name');
```

3. Liest alle Daten im Tab und kopiert den gesamten Inhalt.

```
var SRange = sheet_gsc_with.getDataRange();
var A1Range = SRange.getA1Notation();
var SData = SRange.getValues();
```

4. Öffnet die Google-Tabelle und den Tab, in dem die Daten gespeichert werden sollen.

```
var target_gsc_with = SpreadsheetApp.openById(,Tabellen-ID');
var tsheet_gsc_with = target_gsc_with.getSheetByName(,Tab-Name');
```

5. Springt zur Zeile nach der letzten Zeile, die Daten beinhaltet.

```
var lastRow = tsheet_gsc_with.getLastRow() + 1;
```

6. Fügt die kopierten Daten ein.

```
var range = tsheet_gsc_with.getRange(lastRow,1,SData.length,SData[0].length);
range.setValues(SData);
```

7. Löscht die Zeile mit der Headline aus der Quell-Datei.

```
tsheet_gsc_with.deleteRow(lastRow);
```

Die Google-Analytics-Daten werden mit einer Zusammenfassung am Anfang der Tabelle ausgeliefert, weshalb das Skript bei den Analytics-Daten die ersten 14 Zeilen im Quell-Tab herauslöscht. Beim nächsten Back-up werden diese Daten dem Tab wieder hinzugefügt.

```
var source_ga = SpreadsheetApp.openById(,Tabellen-ID');
var sheet_ga = source_ga.getSheetByName(,last1day');
source_ga.deleteRows(1,14) // löscht Zeilen 1-14 in ,last1day'
var SRange = sheet_gsc_with.getDataRange();
var A1Range = SRange.getA1Notation();
var SData = SRange.getValues();
var target_ga = SpreadsheetApp.openById(,Tabellen-ID');
var tsheet_ga = target_ga.getSheetByName(,Tab-Name');
var lastRow = tsheet_gsc_with.getLastRow() + 1;
var range = tsheet_gsc_with.getRange(lastRow,1,SData.length,SData[0].length);
range.setValues(SData);
tsheet_gsc_with.deleteRow(lastRow);
```

Eingebunden wird das Skript in diesem SEO-DWH-Beispiel am Ende in der Reporting-Zusammenfassung. Es kann über den Menüpunkt „Tools > Skripteditor ...“ in die Google-Tabelle eingetragen werden. Bindet man jetzt ein Bild in den entsprechenden Tab ein, kann das Skript mit einem Klick auf das Bild über das Bild-Menü „Skript zuweisen“ auf das Bild gelegt werden. Ein nächster Klick auf das Bild führt das Skript dann aus.

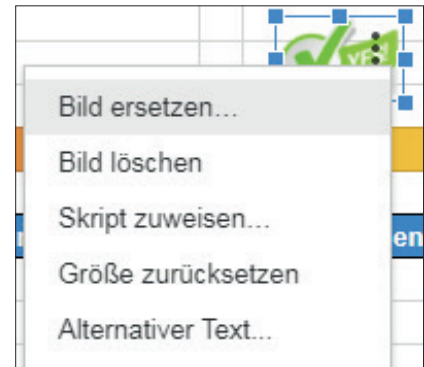


Abb.6: Google Apps Script einem Bild zuweisen

Man sollte immer wieder in die Back-up-Dateien reinschauen. Zum einen können Google-Tabellen, wie eingangs bereits erwähnt, nur zwei Millionen Zellen abspeichern. Zum anderen werden die Dateien irgendwann sehr groß, was die tägliche Analyse erschweren kann.

Es ist ratsam, in den „back-up“-Dateien mindestens die letzten 35 Tage zu behalten und alles andere in neue Dateien in einem Archiv-Ordner abzulegen. Zur besseren Übersicht sind monatliche Back-up-Dateien eine gute Wahl. Diese Dateien können dann später für historische Reportings heruntergeladen werden.

Schritt 3: Daten zusammenfügen

Für ein ausführliches URL-Reporting sollte man beide Datensätze in einem Reporting miteinander verbinden. Dafür erstellt man eine neue Datei. Hier wird sie im „data“-Ordner unter dem Namen „raw_data“ abgelegt.

In der Datei „raw_data“ werden drei Tabs erstellt: ein Tab für die Datensätze aus Analytics „GA_daily“, ein Tab für die Datensätze aus der Search Console ohne Keywords „GSC_w/o_query_daily“ und ein Tab für eine Tabelle, in der beide Quellen miteinander verbunden werden: „combined_data“.

Für das tägliche Reporting sollten in den zwei Tabs „GSC_w/o_query_daily“ und „GA_daily“ mindestens so viele Zeilen hinzugefügt werden, wie ca. 60 Tage Daten benötigen. Wie viele Zeilen das sind, kann man sich leicht ausrechnen,

in dem man die Anzahl aller seiner URLs mit drei (ein Eintrag pro Device) multipliziert und das für 60 Tage hochrechnet. Legt man weniger Zeilen an, kann es passieren, dass nicht alle Daten aus den Back-up-Dateien übertragen werden können.

Schritt 3.1: Daten aus der Search Console aufbereiten

Die Datensätze aus Search Console und Analytics lassen sich direkt aus den Back-up-Dateien laden. Mit der Funktion: =importrange(„https://docs.google.com/spreadsheets/d/[Table-ID]/“, „[Tab-Name]!A2:I“) lassen sich Daten aus einer Datei/einem Tab in eine andere Datei oder einen anderen Tab übertragen. Dabei werden die ausgewählten Daten in die Ziel-Tabelle gespiegelt und in Echtzeit übertragen, nicht kopiert.

Diese Funktion kann an jeder Stelle eines Tabs geschrieben werden. In diesem Beispiel platzieren wir sie im Tab „GSC_w/o_query_daily“ in die Zelle B2. Die erste Zeile wird für den Tabellenkopf verwendet. Dabei ergeben sich Spalten-Bezeichnungen wie in Tabelle 1 zu sehen.

Worauf man bereits hier achten sollte, ist eine einheitliche Bezeichnung der Landingpages. Also in beiden Tabs „GSC_w/o_query_daily“ und „GA_daily“ sollte es in Spalte A die Liste der URLs mit der gleichen Namensgebung geben. Dabei ist es jedem selbst überlassen, ob er sich für nur den URL-Slug /landingpage/ oder Domain-Namen plus URL-Slug „domain.tld/landingpage/“ entscheidet oder ob er Protokoll und Subdomain dabei haben möchte „https://www.domain.tld/landingpage/“.

In diesem Beispiel werden die Spalten B bis H direkt aus der Datei „backup GSC w/o Query“ geladen, wohingegen

A	B	C	D	E	F	G	H	I
URL	Datum	Seite	Device	Klicks	Impressionen	CTR	Position	Pos*Imp

Tabelle 1

A	B	C	D	E	F	G	H	I
URL	Device	Datum	Nutzer	Neue Nutzer	Sitzungen	Absprünge	Sitzungsdauer	Besuchte Seiten

Tabelle 1

Spalte A „URL“ mittels RECHTS-Funktion auf den URL-Slug gekürzt wird. Die Funktion lautet: =WENNFEHLER(RECHTS(C2,LÄNGE(C2)-[XX]),““) und besagt:

„Zeige alle Zeichen auf der rechten Seite der Zelle an, wenn du von links [XX] Zeichen abziehst.“ [XX] ist die Anzahl der Zeichen für Protokoll + Subdomain + domain + Top-Level-Domain. Bei https://www.domain.tld wären das 22 Zeichen.

Die Formel wird bereits zu Beginn auf alle vorhandenen Zeilen angewendet. Um dann nicht eine ganze Liste an #NV-Einträgen in der URL-Spalte zu haben, wurde hier die WENNFEHLER-Funktion genutzt, die leere Einträge hinterlässt. Damit verhindert man, dass jetzt noch nicht übertragende Daten Fehler auslösen.

In der Spalte H „Pos*Imp“ werden die Positionen der Zeile mit den dazugehörigen Impressions multipliziert. Da die Datensätze später zusammengefasst werden, würde bei einer einfachen Durchschnitts-Berechnung eine falsche Position errechnet werden.

Beispiel:

Keyword A rankt auf Position 1 und Keyword B auf Position 3. Eine Durchschnitts-Berechnung der Position würde 2 ergeben. Keyword A hat jedoch 5.000 Impressions, wohingegen

Keyword B nur 1.000 Impressions hat. Somit lautet die Berechnung (5.000*1) + (1.000*3) = 8.000 / 6.000 Impressions, was am Ende eine gewichtete Position von 1,3 ergibt.

Schritt 3.2: Daten aus Analytics aufbereiten

Je nachdem, wie die Landingpages in Analytics bezeichnet werden, müssen sie nicht erneut bearbeitet werden. Da in diesem Beispiel nur der URL-Slug in Analytics ausgegeben wird, muss hier keine extra Spalte hinzugefügt werden. Somit sieht der Tabellenkopf wie in Tabelle 2 aus.

Schritt 3.3: Auswertungs-Tabelle erstellen

Im Tab „combined_data“ wird eine Tabelle mit Spalten, wie in Tabelle 3 zu sehen, erstellt.

Die Spalte „Cluster“ ist optional und dient dazu, bestimmte Teilbereiche einer Domain besser analysieren zu können.

Der nun folgende Teil klingt etwas kompliziert, ist aber recht einfach. Um die Tabelle dauerhaft nutzen zu können, müssen wir zunächst eine Liste aller URLs der Domain für jeden Device-Typ anlegen. Diese werden dann für die Anzahl der Tage, die man einsehen möchte, angelegt.

In diesem Beispiel betrachten wir die Daten der letzten fünf Wochen, das

A	B	C	D	E	F	G	H	I	J	K	L	M	N
URL	Cluster	Device	Datum	KW	Nutzer	Neue Nutzer	Sitzungen	Absprünge	Sitzungsdauer	Besuchte Seiten	Impressionen	Klicks	Pos*Imp

Tabelle 3

heißt, wir müssen im Report Datensätze für 35 Tage bereitstellen.

Zunächst stellt man sich eine Liste aller zu analysierenden URLs einer Domain zusammen. Werden Cluster verwendet, können diese hier schon zugeordnet werden. Diese Liste kopiert man dreimal untereinander und ordnet jedem Item einmal Desktop, Mobile und Tablet zu. Die Liste, die dann entsteht, finden Sie in Abbildung 7.

Um jetzt jede URL mit jedem Device-Typ jeden Tag analysieren zu können, wird diese Liste 35-mal untereinander kopiert. Jetzt ist es wichtig, dass diese Liste nicht mehr umsortiert wird, da die Datums-Zuordnung mittels WENN-Funktion angelegt wird.

Um einen vollständig automatisierten Report zu haben und das Datum nicht jeden Tag händisch eingtragen zu müssen, wird hier eine Formel zu Hilfe genommen. In Zelle D2 (erstes Datum in der Tabelle) wird die Formel „=heute()“ eingetragen. Damit wird hier jeden Tag das aktuelle Datum stehen. In Zelle D3 wird dann folgende WENN-Funktion platziert: =WENN(UND(A3=“/“,C3=“desktop“),D2-1,D2).

Die Funktion überprüft, ob in der jeweiligen Zeile als URL die Homepage und als Device „desktop“ eingetragen sind. Ist das der Fall, wird vom vorhergehenden Datum ein Tag abgezogen. Ist das nicht der Fall, wird das gleiche Datum wie in der Zelle darüber eingetragen. Somit wird jeder Datensatz mit einem Datum versehen und es entstehen keine Duplikate.

Um die Wochenperformance der einzelnen URLs in das tägliche Reporting zu integrieren, sollte man auch die Kalenderwoche zum Datum hinzufügen. Dies kann man zum Beispiel mit einem extra Tab tun, in den man einen Kalender mit Datum und Kalenderwoche einfügt und sich dann per SVERWEIS die Wochennummer zum jeweiligen Datum ausgeben lässt.

URL	Cluster	device
/	home	desktop
/url-1/	cluster 1	desktop
/url-2/	cluster 2	desktop
/url-3/	cluster 1	desktop
/	home	mobile
/url-1/	cluster 1	mobile
/url-2/	cluster 2	mobile
/url-3/	cluster 1	mobile
/	home	tablet
/url-1/	cluster 1	tablet
/url-2/	cluster 2	tablet
/url-3/	cluster 1	tablet

Abb.7: Liste der URLs mit Clustern und Devices

Abb.8: Pivot-Tabelle „pivot_CW“ inklusive Sidebar

Alle anderen Daten werden per SUMMEWENNS-Funktion aus den Analytics- und Search-Console-Datensätzen in den „combined_data“-Tab geladen.

Schritt 4: Pivot-Tabellen erstellen

Wenn man jetzt eine große Tabelle mit allen benötigten Daten zusammengetragen hat, kann man daraus jeweils eine Pivot-Tabelle für die tägliche Domain-Zusammenfassung und eine auf Wochenbasis erstellen. Dafür kann man sich entscheiden, ob man die Pivot-Tabellen in der gleichen Tabelle wie die „combined_data“-Tabelle bauen möchte oder ob man dafür lieber eine extra Datei anlegen will. Hier wird zur besseren Übersicht eine neue Datei „Zusammenfassung“ verwendet.

Da es in Google-Tabellen nur dann möglich ist, eine Pivot-Tabelle zu erstellen, wenn die Rohdaten in der gleichen Datei liegen, muss man sich den „combined_data“-Tab wieder mit der IMPORTRANGE-Funktion in das neue Sheet laden. Daraufhin erstellt man die

Pivot-Tabelle über „Daten > Pivot-Tabelle“.

Es wird ein neuer Tab erstellt und geöffnet. Der Tab wird als „pivot_CW“ benannt. Auf der rechten Seite befindet sich jetzt eine Sidebar, die anzeigt, welche Daten für die Pivot-Tabelle ausgewählt sind. Darunter kann man „Zeilen“, „Spalten“, „Werte“ und „Filter“ hinzufügen. In „Zeilen“ sollte man sich die Zeiträume anzeigen lassen, also die Kalenderwoche. Alles Weitere wird unter „Werte“ eingefügt.

Sitzungen, Impressionen und Klicks können ganz normal über das Dropdown-Menü ausgewählt werden. Alles andere muss als „Berechnetes Feld“ über Formeln erstellt werden - siehe Tabelle 4.

Warum wird hier die WENNFEHLER-Funktion verwendet? Bei vielen dieser Werte kann es sein, dass sie nicht vorhanden sind. Dann zeigt das System zum Beispiel „#DIV/0!“ an. Also einen Fehler, dass hier durch 0 geteilt wurde. Dadurch kann es dazu kommen,

New-User-Rate	=WENNFEHLER(,newUsers/'sessions';0)
Bounce-Rate	=WENNFEHLER(,bounces/'sessions';0)
avg. TpV*	=WENNFEHLER(,sessionDuration/'sessions')/86400;0)
avg. PpV**	=WENNFEHLER(,pageviews/'sessions';0)
CTR	=WENNFEHLER(Clicks/Impressions;0)
avg. Pos.	=WENNFEHLER(,Pos*Imp/Impressions;0)
* (average Time per Visit) // ** (average Pages per Visit)	

Tabelle 4

Abb.9: SEO-Dashboard

dass in der Zusammenfassung keine Werte angezeigt werden. Um das zu verhindern, wird hier mit der WENNFEHLER-Funktion eine 0 eingetragen.

Bei der Berechnung der durchschnittlichen Sitzungsdauer wird der Quotient noch mal durch 86.400 geteilt. Warum? Analytics gibt die Sitzungsdauer in Sekunden aus. Will man jedoch die Dauer im Format HH:MM:SS angezeigt bekommen, muss man die Anzahl der Sekunden durch das Produkt von 24(Stunden)*60(Minuten)*60(Sekunden), also 86.400, teilen.

Als Letztes werden nun noch die Spalten richtig formatiert. Das heißt „Rate neue Nutzer“, „Absprungrate“ und „CTR“ sind Prozentwerte. „Ø Sitzungsdauer“ wird als Dauer in HH:MM:SS angezeigt. „Ø Seiten/Sitzung“ und „gew. Position“ sollten mindestens eine bis zwei Nachkommastellen zeigen.

Die zweite Pivot-Tabelle „pivot_daily“ kann einfach erstellt werden, indem man die „pivot_CW“ mit einem Rechtsklick auf den Tab-Namen dupliziert und in „Spalten“ „KW“ löscht und „Datum“ einfügt.

Schritt 5: SEO-Dashboard erstellen

Im letzten Schritt erstellt man sich einen neuen Tab „Übersicht“, in dem die zusammengestellten Daten auf einen Blick einsehbar sind.

Für die tägliche Zusammenfassung wurde hier eine Übersicht der letzten acht Tage gewählt, so lässt sich abschätzen, ob der vorherige Tag gleich, besser oder schlechter als der gleiche Tag eine Woche zuvor verlaufen ist. In der Wochenübersicht sieht man hier eine Zusammenfassung der aktuellen plus der vier Wochen zuvor, um einen Trend erkennen zu können.

Zusätzlich betrachten wir noch den gleichen Wochentag aus der aktuellen Woche und den vier Wochen zuvor. Ist der 8. Januar 2018 zum Beispiel ein Montag, lässt sich abschätzen, ob es im Vergleich ein „guter“ Montag ist oder eher nicht.

Zunächst wird in Zelle B wieder die HEUTE-Funktion eingetragen, um immer das aktuelle Datum ausgegeben zu bekommen. Dann werden die Daten-Tabellen erstellt. Das Datum in

Zelle A7 lässt sich einfach durch =A2-1 eintragen. Auch die anderen Daten werden genauso automatisiert, dann aber mit =A7-1, es wird also immer ein Tag von der vorhergehenden Zelle abgezogen. Bei der Wochentags-Analyse werden immer sieben Tage abgezogen.

Alle Werte werden dann mittels SVERWEIS-Funktion aus den Pivot-Tabs in die Zusammenfassung übertragen.

In diesem Beispiel wurde das „Google Apps Script“ in den „Übersicht“-Tab eingefügt und auf ein Bild angewendet. Ein Linksklick auf dieses Bild führt das Skript aus. Bis alle Daten geladen sind, kann es jedoch ein paar Sekunden bis zu einer Minute dauern.

Zusatz: Sistrx einbinden

Über die IMPORTXML-Funktion ist es möglich, in Google-Tabellen auch in XML ausgegebene Custom APIs einzubinden. Sistrx zum Beispiel verwendet dieses Format. Damit lassen sich auch alle Domain-Daten in den täglichen Report mit einbinden. Der erste Wert der Funktion ist die entsprechende Sistrx-API-URL, der zweite Wert lautet „//@value“.

Fazit

Sich ein Basis-SEO-DWH zu bauen, ist damit in ein bis zwei Tagen möglich. Gerade kleinere Projekte können von einem SEO-DWH mit Google-Tabellen profitieren. Die Möglichkeit, mit nur wenigen Klicks eine gesamte Domain analysieren zu können, spart viel Zeit. Auch ein tägliches Back-up aller relevanten Daten ist für eine historische Betrachtung einer Domain unverzichtbar.

Dennoch haben Google-Tabellen einige Limitierungen und sind für große Projekte weniger geeignet. Wie man ein SEO-DWH für große Projekte erstellt, lesen Sie in der nächsten Ausgabe. ¶