

Katharina Koch

SEO-Data-Warehousing (Teil 2 – für große Projekte mit SQL-Datenbanken)

In der letzten Ausgabe wurde ausführlich darüber berichtet, wie mittels Google Sheets ein eigenes SEO-Data-Warehouse aufgebaut werden kann. Die Lösung ist für kleine bis mittlere Projekte gut geeignet, betreut man jedoch ein größeres Projekt, kommt man mit Google Sheets meist schnell an ein Limit. Deshalb ist es in solchen Fällen ratsam, eine Datenbank-Lösung zu nutzen. Dieser zweite Teil befasst sich daher mit dem Aufbau solcher Lösungen und damit, welche Vor- und Nachteile sie bieten.

Bevor man anfängt, eine SEO-Datenbank aufzubauen, sollten einige Fragen geklärt werden:

- » Was möchte ich mit meinem Data-Warehouse eigentlich machen?
- » Wie viel Geld möchte und kann ich für diese Lösung ausgeben?
- » Sollen die Daten auf einem Server gehostet werden oder lieber auf einem stationären Rechner liegen?
- » Sollen nur Daten gesammelt werden oder die Möglichkeit, die Daten über ein Tool auswerten zu können?

Dieser Artikel erläutert, wie man Daten aus Google Analytics und der Google Search Console in eine MySQL-Datenbank laden kann. Dabei wird auf den Unterschied zwischen einer Datenbank auf einem Server und einer lokalen Datenbank auf dem eigenen PC eingegangen.

Schritt 1: Datenbank-Struktur festlegen

Bevor man über eine Infrastruktur nachdenkt, sollte man überlegen, wie die Datenbank und ihre Tabellen aussehen sollen.

Der Einfachheit halber werden hier nur Basis-Daten aus den bekanntesten Google-Tools gespeichert: aus Google Analytics Daten aus dem Zielseiten-Bericht, dazu gehören die KPIs: Sitzungen,

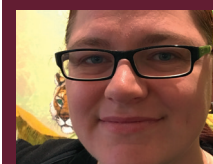
Nutzer, neue User, Absprünge, Seiten pro Sitzung und Sitzungsdauer. Als primäre Dimensionen werden die Zielseiten-URLs und das Datum verwendet. Aus der Google Search Console werden Impressionen, Klicks, durchschnittliche CTR und durchschnittliche Position, einmal auf URL-Level und einmal auf Suchanfragen-Level mit URLs für jedes Datum heruntergeladen und gespeichert. Da Mobile einen immer größeren Teil des organischen Traffics ausmacht, sollten die Daten nach Geräte-Typ aufgeteilt werden.

Sollen in der späteren Analyse andere Daten wie zum Beispiel Conversion-Daten hinzugezogen werden, so können die entsprechenden Tabellen ergänzt werden.

Wichtig ist, dass pro Quelle und Berichts-Typ eine Tabelle innerhalb der Datenbank angelegt wird. Möchte man aus Google Analytics sowohl den Zielseiten-Report als auch den Kampagnen-Report speichern, wird eine Tabelle für die Daten aus dem Zielseiten-Report und eine für die Kampagnen-Daten angelegt, da beide Berichte unterschiedliche Dimensionen als Datenbasis aufweisen.

Eine Übersicht der in diesem Artikel genutzten Datenbank und der enthaltenen Tabellen findet sich in Abbildung 1.

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.

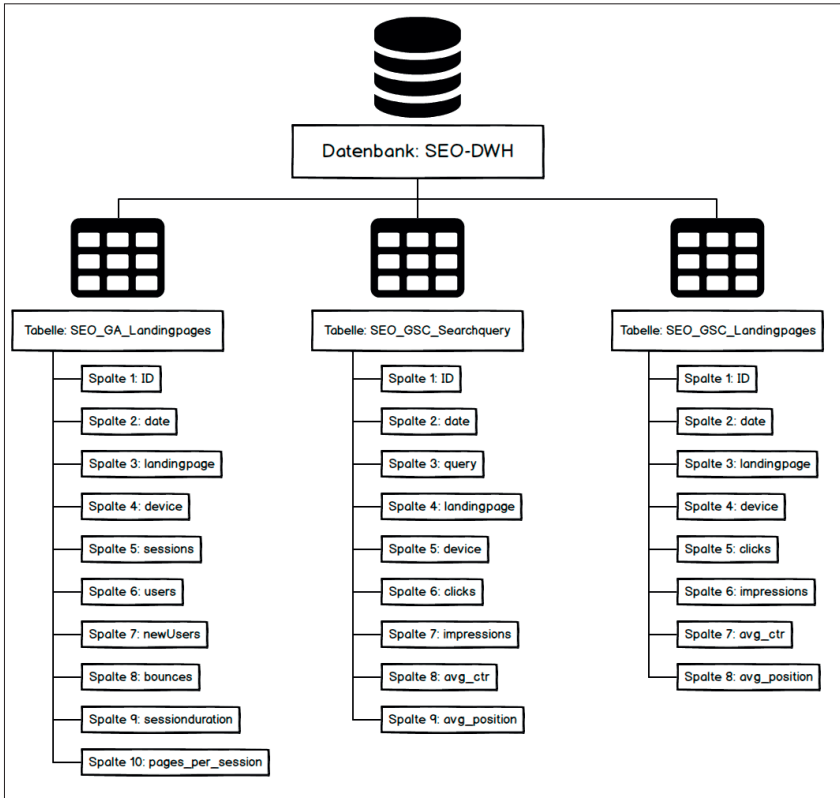


Abb. 1: Die Datenbank-Struktur

Schritt 2: Datenbank-Server mieten oder XAMPP nutzen

Wenn die gewünschte Datenstruktur feststeht, kommt die Frage nach dem Server-Standort. Sollen die Daten auf einem Server liegen oder reicht es, sie auf dem eigenen PC abzuspeichern? Entschieden man sich für eine lokale Lösung, ist XAMPP die einfachste Möglichkeit, eine Datenbank einzurichten. Der Nachteil: Daten können nur dann 100%ig automatisiert heruntergeladen werden, wenn der Rechner an und XAMPP aktiviert ist. Nutzt man einen Server, so kann man sich leicht einen Cronjob einrichten, der den Datendownload zu einer bestimmten Uhrzeit selbst übernimmt.

Worauf achten bei Datenbank-Servern?

Soll die Datenspeicherung zu 100 % automatisiert erfolgen, sollte man das Geld für ein Server-Hosting investieren. Anbieter gibt es viele auf dem Markt, ein paar Sachen gilt es allerdings zu beachten:

- » Min. 1 GB Ram
- » Zurverfügungstellung eines Daten-

bank-Administrations-Tool (z. B. phpMyAdmin)

- » Zugriff per SSH
- » Python-Installation auf dem Server (oder Rechte zur Installation von Python)
- » Möglichkeiten, Cronjobs auszuführen

Die Einrichtung eines Servers sollte man jemandem überlassen, der sich damit auskennt, da hier einiges an technischem Know-how notwendig ist. Es gibt aber auch zahlreiche YouTube-Videos, die dabei helfen. Der Server sollte aber gut abgesichert werden, da die Gefahr, dass er gehackt wird, sonst recht hoch ist.

Im Weiteren wird hier die Einrichtung der Datenbanken mit XAMPP erläutert, da das schnell und ohne große Vorkenntnisse machbar ist.

XAMPP installieren

XAMPP ist ein Server, den man auf seinem eigenen Rechner installieren kann. Damit kann man neue Webseiten testen und aufsetzen, ohne sie gleich im Internet präsentieren zu müssen, und man hat die Möglichkeit, sich mittels

phpMyAdmin eigene Datenbanken anzulegen.

Die Installation von XAMPP ist recht einfach, sucht man im Internet nach „XAMPP download“, findet man schnell viele Gratis-Downloads des Programms.

Zur Verwendung von XAMPP sei noch Folgendes gesagt: Das Programm verwendet die Ports 80 und 443, was zu Konflikten mit anderen Programmen, zum Beispiel Skype, führen kann. Das Internet bietet aber zahlreiche Tutorials, wie diese Probleme gelöst werden können.

Nach der XAMPP-Installation öffnet sich das XAMPP Control Panel. Von hieraus können die einzelnen Teile des Programms aktiviert oder gestoppt werden. Für die Einrichtung von Datenbanken sollten die Module „Apache“ und „MySQL“ gestartet sein (siehe Abbildung 2).

Schritt 3: Datenbank einrichten

Datenbanken mit XAMPP einzurichten, ist im Grunde recht einfach. Unter folgendem Link öffnet sich das phpMyAdmin-Tool in Ihrem Browser: „localhost/phpmyadmin/“ (siehe Abbildung 3).

Schritt 3.1: Datenbank anlegen

In der Sidebar kann man die bereits vorhandenen Datenbanken sehen. Unter dem Punkt „neu“ lassen sich neue Datenbanken anlegen. Dazu reicht es, der Datenbank einfach einen Namen zu geben und als Format „Kollation“ auszuwählen. Alles andere wird in den Tabellen selbst eingerichtet. Über den „Anlegen“-Button wird die neue Datenbank angelegt.

Schritt 3.2: Tabellen anlegen

Um die benötigten Tabellen anzulegen, reicht ein CSV-Upload. Die ersten Datensätze kann man sich, wie im letzten Artikel beschrieben, einfach

mit den Google Sheets und den Add-ons heruntergeladen. Über „Datei > Herunterladen als > Kommagetrennte Werte (.csv aktuelles Tabellenblatt)“ kann die CSV-Datei heruntergeladen werden.

Da jeder Datenbank-Eintrag ein einmaliges Identifier benötigt, wird der Tabelle eine Extra-Spalte mit dem Spalten-Namen „IDs“ hinzugefügt. Jeder Eintrag wird dann mit einer fortlaufenden Nummer besetzt. Dabei ist es egal, wo man zu zählen anfängt.

Zurück im Browser kann über den Button „Datei auswählen“ die CSV-Datei hochgeladen werden. Als Zeichencodierung sollte „utf-8“ verwendet werden. Das Komma dient als Spaltentrenner, Spalten werden von Anführungszeichen eingeschlossen und beendet. Man sollte darauf achten, dass das Häkchen vor der Information „Die erste Zeile der Datei enthält die Spaltennamen (...)“ aktiviert ist, andernfalls werden die Spaltenbeschriftungen als Daten interpretiert, was das Reporting behindern kann.

Ein Klick auf den OK-Button legt eine neue Tabelle mit dem Namen „tabelle 1“ an. Über den Reiter „Operationen > Tabellenoptionen“ kann der Tabelle eine sinnvolle Bezeichnung gegeben werden.

Schritt 3.3: Tabellen formatieren

Dann müssen die einzelnen Tabellen-Spalten richtig formatiert werden.

Unter dem Reiter „Struktur“ lassen sich die Formatierungen einstellen. Dafür auf den „Bearbeiten“-Button der jeweiligen Spalte klicken. In dem sich öffnenden Tab werden Spaltenname, Typ, Länge/Werte u. a. eingestellt.

Für die benötigten Spalten sollten folgende Einstellungen vorgenommen werden:

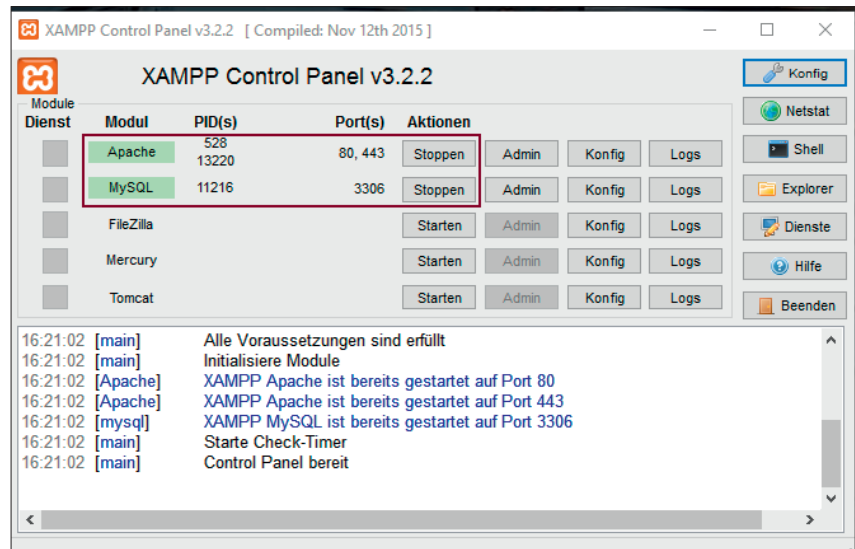


Abb. 2: XAMPP Control Panel

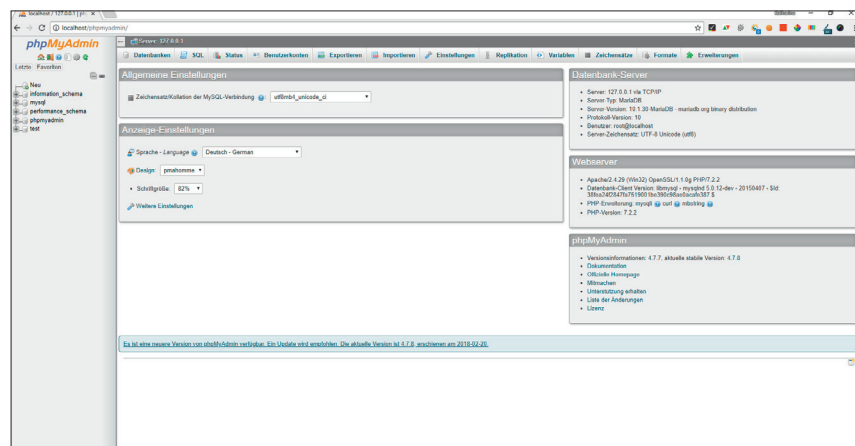


Abb. 3: Das phpMyAdmin-Dashboard

- » **ID**
 - Typ: VARCHAR
 - Länge: 5 oder 6 (bei der Länge kommt es darauf an, wie lang die ID werden kann/soll)
- » **landingpage**
 - Typ: VARCHAR
 - Länge: 255
 - Kollation: utf_8_general_ci
- » **device**
 - Typ: VARCHAR
 - Länge: 7 – nur 3 Werte möglich: mobile (6 Zeichen), desktop (7 Zeichen), tablet (6 Zeichen)
 - Kollation: utf_8_general_ci
- » **date**
 - Typ: DATE
- » **users, newUsers, pages_per_session, clicks, impressions**

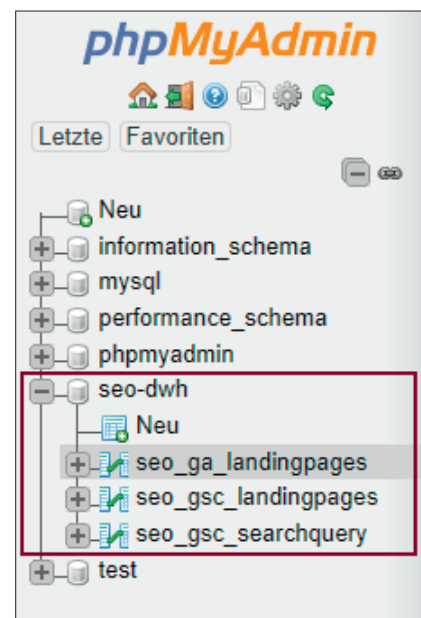


Abb. 4: Datenbank-Struktur SEO-DWH

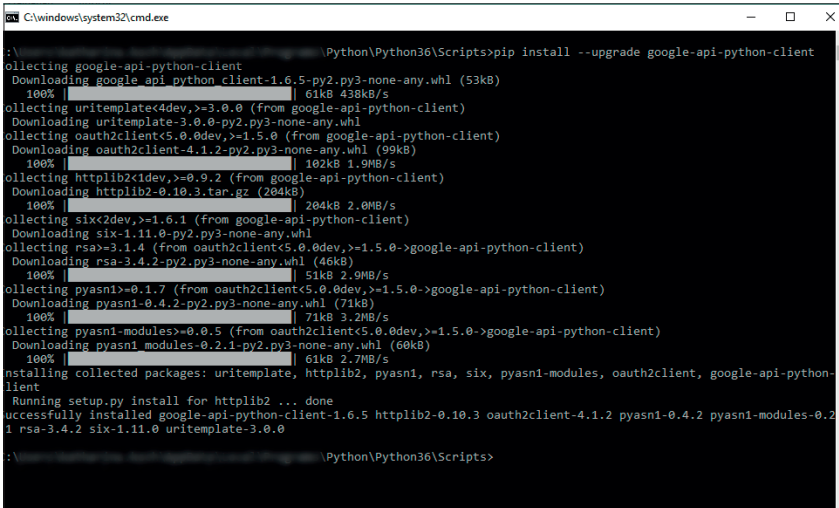


Abb. 5: Google API Python Client in Python installieren

| API | Anfragen | Fehler | Fehlerverhältnis | Latenz, Median | Latenz, 98 % |
|--------------------------------|----------|--------|------------------|----------------|----------------|
| Google Analytics Reporting API | - | - | - | - | - Deaktivieren |
| Google Search Console API | - | - | - | - | - Deaktivieren |

Abb. 6: Google API Python Client in Python installieren

- Typ: INT
- Länge: 4 (die Länge ist abhängig von der Anzahl der Besucher pro Tag, für mehrere Millionen Nutzer auf einer URL sollte die Zahl entsprechend höher sein)
- » **sessions, bounces, sessionduration**
 - Typ: INT
 - Länge: 5
- » **query**
 - Typ: VARCHAR
 - Länge: 255
 - Kollation: utf_8_general_ci
- » **avg_ctr, avg_pos**
 - Typ: DECIMAL
 - Länge: 3,2 (3 = Anzahl der Zeichen vor dem Komma – also max. 100 % oder Position 999; 2 = Anzahl der Zeichen nach dem Komma, es können auch mehr Nachkomma-Stellen gewählt werden, wenn präzisere Ergebnisse gewünscht sind)

Schritt 4: Python herunterladen und installieren

Python auf dem eigenen PC zu installieren, geht genauso einfach, wie XAMPP zu installieren. Sucht man im Internet nach „Python download“ findet man schnell viele Down-

load-Dateien. Hierbei darauf achten, welche Version installiert wird, da sich Python 2 und Python 3 z. T. erheblich unterscheiden. Python ist eine Programmier-Sprache, die recht einfach zu erlernen ist. Viele Firmen und Entwickler verwenden Python für Automatisierungen und Machine-Learning-Projekte. Es ist also keine Zeitverschwendung, sich in die Sprache einzuarbeiten.

Über Python auf Windows muss Folgendes gesagt werden: Die mit Python erstellten Programme müssen über die Windows Command Line ausgeführt werden (einfach im Startmenü nach cmd suchen oder auf einen Ordner, aus dem man heraus ein Python-Skript ausführen möchte, mit gedrückter Shift-Taste einen Rechts-Klick ausführen und im Menü dann „Eingabeaufforderung hier öffnen“ auswählen).

Ein Python-Skript kann entweder über die mitgelieferte Konsole oder in jedem anderen beliebigen Editor geschrieben werden, solange er Dateien vom Typ .py abspeichern kann. Es gibt aber auch Editoren, die speziell auf das Programmieren mit Python ausgelegt sind, zum Beispiel atom.io mit dem Python Package oder PyCharm.

BUCH-TIPPS

Da das Thema „Datenbank selbst anlegen und auswerten“ sehr komplex ist, hier noch ein paar Buch-Tipps, die die Einarbeitung in Python und MySQL erleichtern werden.

PYTHON:

Routineaufgaben mit Python automatisieren: Praktische Programmierlösungen für Einsteiger (Al Sweigart, dpunkt, Level: Einsteiger)

Python 3: Das umfassende Handbuch: Sprachgrundlagen, Objektorientierung, Modularisierung (Johannes Ernesti & Peter Kaiser, Rheinwerk Computing, Level: Fortgeschrittene)

MYSQL:

Einstieg in SQL: Für alle wichtigen Datenbanksysteme: MySQL, PostgreSQL, MariaDB, MS SQL. Ohne Vorkenntnisse einsteigen! (Michael Laube, Rheinwerk Computing, Level: Einsteiger)

MySQL: Das umfassende Handbuch (Wolfgang Gassler, Stefan Pröll & Eva Zangerle, Rheinwerk Computing, Level: Fortgeschrittene)

Schritt 5: Google Libraries in Python installieren

Externe Libraries werden unter Windows über die Command Line zu Python hinzugefügt. Um die Google APIs über Python steuern zu können, müssen die Google Libraries installiert werden.

Dazu wird die Command Line mit dem Python-Skript-Ordner geöffnet. Sollte nicht klar sein, wo sich dieser befindet, so kann man im Windows Explorer nach „pip.exe“ suchen.

Der Google API Python Client wird mit folgendem Befehl installiert: pip install –upgrade google-api-python-client (siehe Abb. 5).

Schritt 6: OAuth 2.0 Client-ID anlegen

Bevor man anfangen kann, sein Skript anzulegen und auszuführen, benötigt man zunächst eine OAuth 2.0 Client-ID von Google. Diese kann über folgenden Link eingerichtet werden:

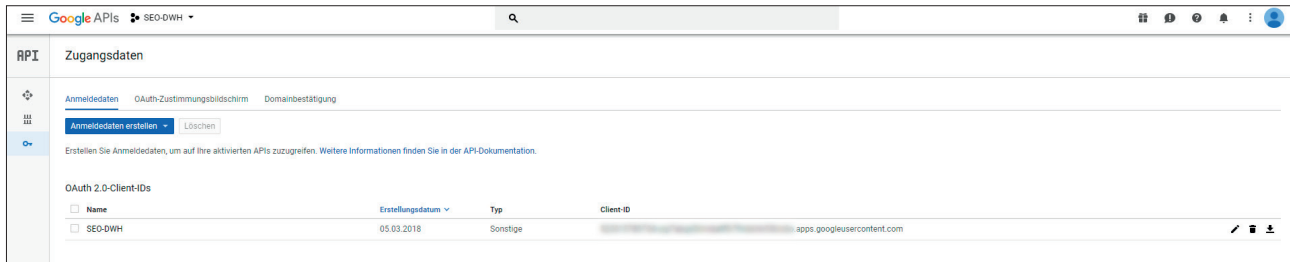


Abb. 7: OAuth 2.0 Client-ID-Liste

<https://console.developers.google.com/apis/dashboard>

Über den Button „+ APIS UND DIENSTE AKTIVIEREN“ können Zugänge für jede Google-API angelegt werden. Im ersten Schritt wird ein neues Projekt angelegt. Dann aktiviert man die APIs der einzelnen Google-Produkte über die Bibliothek (siehe Abbildung 6).

Unter dem Punkt „Zugangsdaten in der Sidebar“ werden die Anmelde-daten erstellt. Dazu auf „Anmelde-daten erstellen“ gehen und den Unterpunkt OAuth-Client-ID anwählen. Als Anwendungstyp „Sonstige“ anklicken. Der Name kann frei gewählt werden. Im Anschluss öffnet sich ein Pop-up, in dem die Client-ID sowie der Client-Schlüssel angezeigt werden. Beides muss nicht extra gespeichert werden, da sowohl die Client-ID als auch der Schlüssel über den Bearbeiten-Button in der OAuth 2.0 Client-ID-Liste angezeigt werden (siehe Abbildung 7).

Schritt 7: Code schreiben

Das Skript für den Datendownload ist der schwierigste Punkt in diesem Projekt. Das Internet bietet viele ausführliche Skripte, die helfen, Python mit den Google APIs zu verbinden.

Gute Basis-Skripte hat Google bereits im Jahr 2015 selbst angelegt. Unter der URL: <https://github.com/google/google-api-python-client/tree/master/samples/> sind Skripte für fast alle Google APIs zu finden. Um die Skripte zu verstehen, sollte man sich schon etwas mit Python beschäftigt haben, da viele Funktionen vorkommen, die im Basis-Wissen nicht oder kaum verwendet werden.

```
{
  „installed“: {
    „client_id“: „[client ID - endet mit: .apps.googleusercontent.com]“,
    „client_secret“: „[client-secret-code]“,
    „redirect_uris“: [„urn:iETF:wg:oauth:2.0:oob“, „http://localhost“],
    „auth_uri“: „https://accounts.google.com/o/oauth2/auth“,
    „token_uri“: „https://accounts.google.com/o/oauth2/token“
  }
}
```

Code 1

Wichtig ist die client_secrets.json-Datei. Hier werden die Zugangs-Informationen zur API gespeichert. Diese Zugangsdaten sind die OAuth 2.0 Client-ID und deren Client-Schlüssel.

„client_secrets.json“ sollte am Ende aussehen wie in Code 1.

Unter <https://searchwilderness.com/gwmt-data-python/> findet man ein gutes Skript, welches nicht nur den Datendownload aus der Search Console beinhaltet, sondern auch den Datenbank-Upload im Anschluss.

Sowohl die Google-API-Python-Skripte also auch das auf searchwilderness.com haben einen gemeinsamen „Nachteil“. Beide Skripte sind bereits fast drei Jahre alt und mit aktuellen Python-Versionen zum Teil nicht kompatibel, sodass ein paar Sachen angepasst werden müssen. Zum Beispiel wird die Funktion „print“ in beiden Skripten wie folgt genutzt:

```
print title + ,:‘
```

In der aktuellen Python-Version 3.6 wird die Print-Funktion allerdings wie folgt genutzt:

```
print (title + ,:‘)
```

Schritt 8: Daten auswerten

Zur Auswertung seiner gespeicherten Daten hat man am Ende mehrere Möglichkeiten. Man kann sich die Daten direkt mittels Workbench und einer eigens dafür geschriebenen SQL-Query herunterladen oder man lässt sich jeden Tag mittels Cronjob eine Excel-Datei zuschicken. Es gibt auch zahlreiche Tools, die Daten visuell zu Analyse-zwecken aufbereiten, z. B. Tableau, Qlik Sense, Microsoft Power BI etc.

Fazit

Eine Datenbank-Lösung für das eigene SEO-DWH zu erstellen, erfordert viel Know-how und technischen Sachverstand. Für die Einrichtung sollte man, je nach Kenntnisstand, zwei Tage bis einige Wochen ansetzen. Gerade um das Verständnis für die Python-Skripte zu entwickeln, braucht es einige Einarbeitungszeit. Kennt man sich nicht damit aus, ist auf jeden Fall Geduld gefragt, da man immer wieder auf Probleme und Fehler stößt. Die genannten Basis-Skripte liefern allerdings eine gute Ausgangsbasis. Am Ende hat man aber die Möglichkeit, seine Domain auch in zwei Jahren noch durchgehend analysieren zu können. ¶