Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Jens Peter Urban

Data Warehouse- und Data Lake-Systeme im Kontext von Big Data

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Jens Peter Urban

Data Warehouse- und Data Lake-Systeme im Kontext von Big Data

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Zukunft
Zweitgutachter: Prof. Dr. Schultz

Eingereicht am: 27.08.2018

Jens Peter Urban

Thema der Arbeit
Data Warehouse- und Data Lake-Systeme im Kontext von Big Data

Stichworte
Data Warehouse, Big Data, Data Warehouse-Architektur, Data Lake

Kurzzusammenfassung
In der heutigen Zeit ist es für Unternehmen wichtiger als jemals zuvor gespeicherte Daten und Informationen zu verarbeiten, um aus diesen wertvolle Analysen auszuführen. In diesen Zusammenhang spielt das Data-Warehouse eine entscheidende Rolle, da mit diesen Systemen Prozessoptimierungen, Analysen und Verbesserungen umgesetzt werden können.

Jens Peter Urban

Title of the paper
Data warehouse and data lake systems in the context of big data

Keywords
Data Warehouse, Big Data, Data Warehouse-Architecture, Data Lake

Abstract
Nowadays, it is more important than ever for companies to process stored data and information in order to perform these valuable analyses. The data warehouse plays a decisive role in this context, as these systems can be used to implement process optimizations, analyses and improvements.

Inhaltsverzeichnis

1. Einleitung

In diesem Kapitel wird zur Problemstellung hingeführt. Danach wird die Zielsetzung ausgeführt und genauer erläutert. Abschließend wird der Aufbau der Arbeit beschrieben, um eine einfachere Handhabe zu gewährleisten und die thematische Entwicklung darzustellen.

1.1. Problembeschreibung

Schon früh wurde die Verarbeitung und Speicherung von Daten zum Thema. So entstand das Data Warehouse, das von Inmon folgendermaßen definiert wird: „*A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process* "(Inmon, 2005, S. 33).

In vielen Unternehmen werden Data Warehouse-Systeme benutzt, um Unternehmensdaten zu strukturieren, zu analysieren und auszuwerten. Auf diese Weise können die Daten, die in einem Data Warehouse verarbeitet wurden, genutzt werden. So lassen sich aus den gewonnenen Ergebnissen Prozessoptimierungen ableiten. Ein Data Warehouse integriert Informationen aus vielen unterschiedlichen Quellen in einer für die Entscheidungsfindung optimierten Datenbank (vgl. Gluchowski, 2012).

Im Laufe der Zeit sind die Ansprüche an Data Warehouse-Systeme immer komplexer geworden: Obwohl Unternehmen immer noch klassische Data Warehouse-Systeme benutzen, um ihre Daten auszuwerten und zu verarbeiten (vgl. Bauer und Günzel, 2013), wird es immer schwieriger, die wachsende Menge an Daten schnell und effizient in diesen Systemen zu verarbeiten. Dementsprechend war ein Umdenken erforderlich, um diese neuen Informationen gleichermaßen schnell verarbeiten zu können.

Im Kontext dieser Informationen ist auch von ‚Big Data' die Rede. Der Begriff Big Data bezeichnet Datenmengen, die zu groß, zu schnelllebig oder nicht genug strukturiert sind, um sie mit manuellen und herkömmlichen Methoden der Datenverarbeitung auswerten zu können. Um den Ansprüchen gerecht zu werden, müssen moderne Data Warehouse-Systeme auch

diese Informationen verarbeiten können. In diesem Zusammenhang ist auch von den 3 Vs‘ die Rede: Volume, Variety und Velocity. Diese wichtigen Attribute beschreiben die Datenmengen, die Datenvielfalt und die Datengeschwindigkeit (vgl. Klein u. a., 2013). Um diesen Ansprüchen auch weiterhin gerecht zu werden mussten neue Technologien entwickelt werden. Eine dieser Technologien ist der Data Lake. Dieser kann als Ergänzung zu einem Data Warehouse gesehen werden. Der Data Lake ermöglicht eine Verarbeitung großer Datenmengen im Kontext von Big Data.

1.2. Ziel der Arbeit

Ziel der Arbeit ist ein Vergleich von klassischen Data Warehouse-Systemen mit modernen Data Warehouse-Systemen. Im Kontext von Big Data haben sich die Ansprüche an diese Systeme stark verändert und es gibt viele neue Ansätze, die klassische Data Warehouse-Systeme ergänzen sollen. Data Lake-Systeme als Erweiterung in bestehenden Data Warehouse-Umgebungen helfen bei der Bewältigung von neuen Herausforderungen im Kontext von Big Data. Eine Gegenüberstellung der Ansätze zeigt, dass beide Architekturen ihre Vor- und Nachteile haben. Daraus ergibt sich, dass die Entscheidung für eines dieser Systeme immer von den Ansprüchen abhängt, die ein Unternehmen hat. So sollte situationsbedingt abgeschätzt werden, welche Architektur den jeweiligen Bedürfnissen am besten entspricht (vgl. Dull, 2015). Viele neue Technologien in diesem Bereich ermöglichen eine effizientere Verarbeitung größerer Datenmengen. Vor allem im Cloud-Bereich gibt es verschiedene Lösungen, die den Anwendern zur Verfügung stehen. Drei dieser Cloud-Anbieter werden experimentell Untersucht, dabei soll der Fokus auf Teilkomponenten von Data Warehouse- und Data Lake-Systemen gelegt werden. Die Realisierung soll zeigen, wie mit Cloud-Anbietern diese Systeme umgesetzt werden können. Weiterhin wird ein Vergleich der Cloud-Dienste aufgezeigt.

1.3. Struktur der Arbeit

Das erste Kapitel führt in das Thema der Bachelorarbeit ein: Das Hauptziel, Teilziele und der Aufbau der Arbeit werden dargelegt. In Kapitel 2 werden die technischen Grundlagen von klassischen Data Warehouse-Systemen erläutert, die als Basis dieser Arbeit zu betrachten sind. Dabei werden unter anderem der Extract-Transform-Load-Prozess (ETL-Prozess), die Data Warehouse-Architektur sowie das Online Analytic Processing (OLAP) beschrieben. Im Anschluss an die Definition von Data Warehouse-Systemen werden die modernen Data Warehouse-Systeme erläutert. In Kapitel 3 werden die maßgeblichen Aspekte von Big Data

beschrieben. Dazu gehören NoSQL Datenbanken, Data Lake-Systeme und der Unterschied zwischen ETL und ELT. Anschließend werden in Kapitel 4 konzeptionelle Umsetzungen von Data Warehouse-Systemen vorgestellt. Dabei werden drei Systeme mit einander verglichen und deren Teilkomponenten gegenübergestellt. Den Abschluss der Arbeit bildet das Kapitel 5 mit einer Zusammenfassung der Inhalte und einem Fazit.

2. Klassische Data Warehouse-Systeme

In diesem Kapitel wird das klassischen Data Warehouse vorgestellt. Zweck dieser Systeme ist das Aufbereiten von Unternehmensdaten, die für längere Zeit gespeichert sowie für die Mitarbeiter zur Auswertung und Analyse bereitgestellt werden sollen. In Abschnitt 2.1 werden die verschiedenen Einsatzgebiete für Data Warehouse-Systeme beschrieben. Danach wird in Abschnitt 2.2 die Referenzarchitektur für das Data Warehouse erklärt. Im folgenden Abschnitt 2.3 werden dann die Möglichkeiten zum Importieren und Säubern der Daten erläutert. Dabei wird auf den ETL-Prozess näher eingegangen. Abschnitt 2.4 erläutert die verschiedenen Möglichkeiten der Speicherung in einem Data Warehouse. Dabei wird auch auf die interne Verwaltung der Daten durch das Data Warehouse eingegangen. Im Letzten Abschnitt 2.5 werden die Komponenten gezeigt, die es dem Benutzer ermöglichen, mit den Daten zu arbeiten und welche Möglichkeiten es gibt die transformierten Daten zu visualisieren.

2.1. Data Warehouse Einsatzgebiete

Ein Data Warehouse wird von Unternehmen genutzt, um Informationen aus gespeicherten Daten ableiten zu können. Die Daten kommen aus dem operativen Bereich und sollen ausgewertet und visualisiert werden. Die Nutzung eines Data Warehouse durch ein Unternehmen kann verschiedene Gründe haben. Es kann aus Wettbewerbsgründen erstellt werden oder aufgrund bestimmter Vorschriften (vgl. Bauer und Günzel, 2013, S. 14 f.).

In der Wirtschaft finden sich zahlreiche mögliche Einsatzgebiete. Sie reichen von betriebswirtschaftlichen Aufgaben bis hin zu wissenschaftlichen Auswertungen. Die Daten in einen Data Warehouse sollen auch unternehmensweit Vertrauen gewährleisten, auf das sich Anwender verlassen können. Diese Quelle sollte die Einzige sein (Single Source of Truth), auf die sich Anwender bei der Informationsbeschaffung in einem Unternehmen beziehen (vgl. Pang und Szafron, 2014, S. 575). Da bei einem Data Warehouse neben den internen Datenquellen auch externe Daten hinzukommen, bietet es ein gutes Gesamtbild über ein Unternehmen (vgl. Gabriel u. a., 2008, S. 124 f.). Data Warehouse-Systeme können zur Ermittlung von Konsumgütern genutzt werden. Durch das Sammeln von Daten ist es möglich, Rückschlüsse auf das

Kaufverhalten der Konsumenten zu ziehen. Diese Ergebnisse sollen dem Unternehmen dabei helfen, die richtigen Entscheidungen zu treffen. Es kann so beispielsweise darüber entschieden werden, welche Güter weiterhin angeboten werden sollen. Die Ergebnisse geben Aufschluss über die Daten der Kunden und bringen diese in einen sinnvollen Zusammenhang. Weiterhin können auf diese Weise auch Informationen über Alter, Geschlecht, Wohnort und Energieverbrauch ermittelt werden. Data Warehouse-Systeme werden heute in zahlreichen Bereichen der Industrie eingesetzt und sind für viele Betriebe unersetzlich geworden.

2.2. Data Warehouse-Referenzarchitektur

Es gibt verschiedene Ansätze, um ein Data Warehouse in die Praxis umzusetzen. Die Referenzarchitektur zeigt daher nur die grundlegenden Bausteine auf, die in einem Data Warehouse vorhanden sein sollten. Diese Grundbausteine können demnach erweitert und ausgearbeitet werden. In der Regel werden alle Daten in dem Core Data Warehouse gespeichert. Auch eine Speicherung in der Staging-Area ist theoretisch umsetzbar. Auf diese Weise wäre es möglich, bei Fehlern weiterhin auf die Daten zurückgreifen zu können, die im Core Data Warehouse nicht mehr verfügbar sind. Allerdings hat dies einen erhöhten Speicherbedarf und mehr Administrationsaufwand zur Folge (vgl. Gabriel u. a., 2008, S. 132).

In der folgenden Abbildung ist eine Referenzarchitektur mit allen zugehörigen Komponenten dargestellt. Abbildung 2.1 zeigt drei Bereiche, die die Grundpfeiler der Architektur ausmachen. Die Daten werden dabei zunächst aus internen oder externen Quellen extrahiert. Danach beginnt die Datenerfassung. Hierbei werden die Daten erfasst, gesäubert und transformiert. Nachdem alle Daten erfasst wurden, werden diese im Core Data Warehouse gespeichert. Im letzten Schritt werden die aufbereiteten Daten an externe Programme oder Data Marts weitergeleitet.

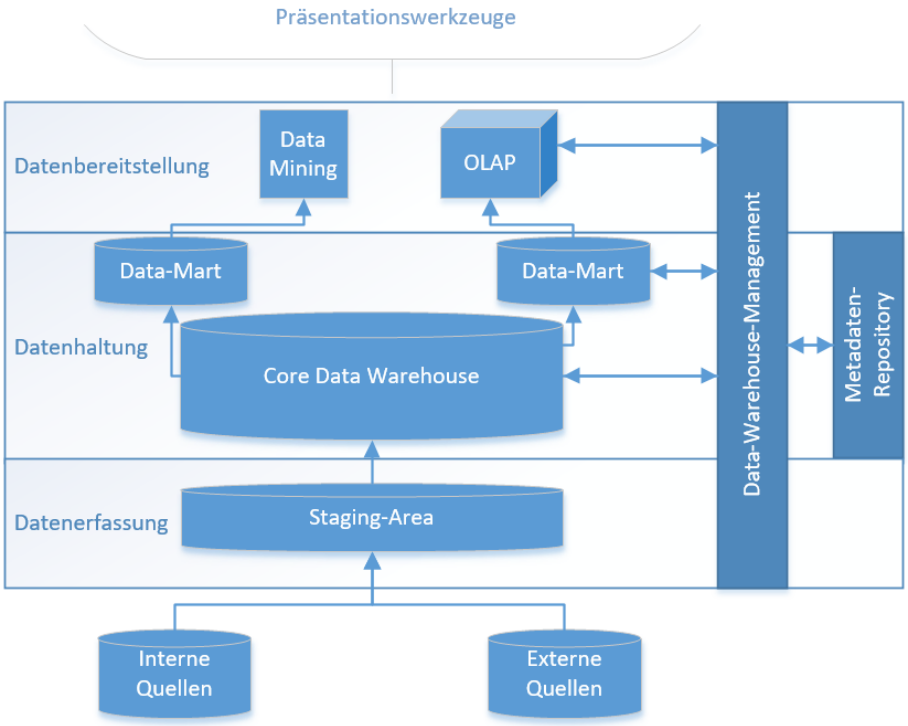


Abbildung 2.1.: Data Warehouse-Referenzarchitektur
Quelle: Modifiziert übernommen aus (Gabriel u. a., 2009, S. 10)

2.3. Importkomponente

Die Daten, die zur Verarbeitung benötigt werden, stammen aus verschiedenen Quellen. Sie sind somit in der Regel heterogen und müssen zur Weiterverarbeitung bereinigt werden. Die Daten können dabei auch aus externen Quellen stammen, wie beispielsweise aus Anwendungen externer Dienstleister, Social Media, Cloud oder Online-Diensten (vgl. Kemper u. a., 2010, S. 28). Stammen die Daten aus internen Quellen, so kommen sie zum Beispiel aus Enterprise-Resource-Planning-Systemen (ERP) oder Excel und liegen in unterschiedlichen Formaten wie CSV oder Textdokumenten vor. Entscheidend ist, dass die Daten zum Befüllen des Data Warehouse-Systems aus einem großen Informationspool kommen, da auf diese Weise eine Analyse von Daten ermöglicht wird, die in einem längeren Zeitraum gesammelt wurden (vgl. Gabriel u. a., 2009, S. 49 ff.). Ein Unternehmen sollte sich darüber im Klaren sein, welche Daten für ein Data Warehouse-System relevant sind. Nicht alle Informationen bringen neue Erkenntnisse oder einen Gewinn für den Unternehmensprozess.

2.3.1. Bereinigung

Bevor die neuen Informationen aus den Daten gewonnen werden können, müssen diese für das Core Data Warehouse aufbereitet werden. Für die weitere Verarbeitung kommen ausschließlich bereits aufbereitete oder gesäuberte Daten infrage. Die Daten werden überprüft und leere oder fehlerhafte Datensätze werden entfernt. Dieser Aufbereitungsvorgang wird als ETL-Prozess bezeichnet. Ziel dabei ist es, die Daten, die aus vielen heterogenen Quellen kommen, für die einzelnen Bereiche einer Organisation bereitzustellen. Nur wenn gesäuberte und organisierte Daten vorliegen, können die Informationen weiterverarbeitet werden (vgl. Hummeltenberg, 2012).

2.3.2. Extract-Transform-Load-Prozess (ETL-Prozess)

In diesem Prozess werden die Daten aus heterogenen Quellen für das Data Warehouse aufbereitet, gespeichert und für die weitere Bearbeitung bereitgestellt. Erst durch den ETL-Prozess ist es möglich, mit Daten effizient und zielführend zu arbeiten. Grundsätzlich ist dabei sicherzustellen, dass die Qualität der Daten erhalten bleibt. Für den Benutzer ist es von Bedeutung, dass die Daten bei der Befüllung aus einem großen Informationspool stammen (vgl. Gabriel u. a., 2008, S. 133).

Im ersten Schritt werden die Daten aus den verschiedenen Quellen extrahiert. Sie werden dann in interne und externe Quellen aufgeteilt. In der Regel wird ein Teil aus der Quelldatei geladen. Dieser Teil wird dann für die Extraktion bereitgestellt. Bei der Extraktion findet eine Schematransformation statt – von dem Schema der Quelle in das Schema des Arbeitsbereiches. Dabei werden die Daten umgeformt und aggregiert. Sie werden dann in das Core Data Warehouse geladen (vgl. Müller und Lenz, 2013, S. 27). Um die Daten in einem Data Warehouse aktuell zu halten, muss in regelmäßigen Abständen eine Extraktion der Unternehmensdaten durchgeführt werden. Nur durch eine regelmäßige Extraktion der Daten kann ein Unternehmen Informationen über einen längeren Zeitraum gewinnen und Vergleiche mit älteren Daten herstellen.

Nachdem die Daten aus verschiedenen Quellen geladen wurden, können diese nun transformiert werden. Die Daten müssen dafür in die Zielstruktur gebracht werden. Die Transformation und Bereinigung der Daten findet in der Staging-Area statt. In diesem Bereich werden die Daten nur temporär in einer SQL-Datenbank gespeichert. Erst wenn die Daten bereinigt wurden,

können diese in das Core Data Warehouse geladen werden.

Zuletzt werden die Daten aus der Staging-Area in das Core Data Warehouse weitergeleitet. Der Ladeprozess soll möglichst schnell und effizient vonstattengehen, da das Zielsystem in dieser Zeit gesperrt ist und folglich nicht damit gearbeitet werden kann. Umgesetzt wird dies dadurch, dass nur diejenigen Bereiche, die geändert werden müssen, auch tatsächlich überschrieben werden (vgl. Bauer und Günzel, 2013, S. 56 ff.). So ist gewährleistet, dass das ganze System nicht vorübergehend unerreichbar ist.

2.4. Verwaltungskomponente

Zentraler Bestandteil des Data Warehouse-Systems ist das Core Data Warehouse. Das Core Data Warehouse besteht in der Regel aus einer relationalen Datenbank. In dieser Datenbank werden alle Daten, die vorher durch ETL-Prozesse aufbereitet wurden, zur Datenhaltung gespeichert. Zudem kommt dem Core Data Warehouse die Aufgabe zu, die Daten zu sammeln und zu integrieren. Auf diese Weise wird die Qualität der Daten gesichert. Vorteil stellen auch die unkomplizierte Pflege und der Betrieb dar, da lediglich eine zentrale Datenbank vorhanden ist (vgl. Chamoni und Gluchowski, 2016, S. 151 f.). Die Daten sollen auf diese Weise über längere Zeiträume archiviert werden und somit eine Historie über viele Jahre bereitstellen. Dies ist etwa bei kundenorientierten Unternehmen notwendig, da diese Daten über den gesamten Lebenszyklus gespeichert bleiben sollen. Die Daten werden dabei meist in mehrdimensionalen Matrizen gespeichert. Das Stern- und das Schneeflockenschema ermöglichen eine relationale Umsetzung der Daten.

2.4.1. Stern- und Schneeflocken-Schema

Eine mögliche Umsetzung von multidimensionalen Konstrukten in ein relationales Datenbankmodell bildet das Sternschema. Dabei handelt es sich – wie auch bei dem Schneeflockenschema – um ein Entitäten-Relationen-Diagramm, eine grafische Darstellung der Tabellen einer Datenbank. Das Sternschema setzt sich grundsätzlich aus einer Faktentabelle und mehreren Dimensionstabellen zusammen. Im Sternschema werden die Tabellen absichtlich denormalisiert gespeichert, um so schnellere Abfragen zu ermöglichen. Die Daten in der Faktentabelle sind normalisiert, aber die Daten in den Dimensionstabellen verstoßen gegen die Normalisierung. Dieser Zustand führt zu Redundanzen, die allerdings erwünscht sind. Beim Sternschema ist lediglich die Faktentabelle mit allen anderen verbunden. Die Dimensionstabellen sind jedoch nicht miteinander verbunden (vgl. Bauer und Günzel, 2013, S. 244 f.). Die beschriebene

Anordnung von Dimensionstabellen und Faktentabelle ist für dieses Schema charakteristisch.

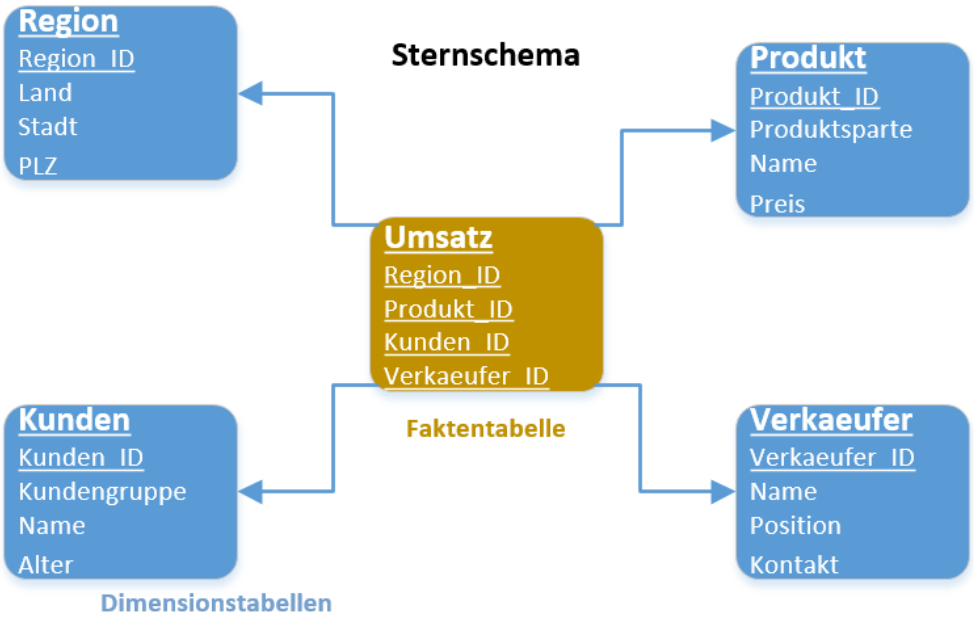


Abbildung 2.2.: Sternschema
Quelle: Modifiziert übernommen aus (Müller und Lenz, 2013, S. 61)

Abbildung 2.2 zeigt, dass der Umsatz in Relation steht zu den Dimensionen Kunden, Region, Verkäufer und Produkt. So lassen sich Abfragen über mehrere Dimensionen erstellen. Eine mögliche Frage könnte etwa lauten: Wie viele Produkte hat ein Verkäufer in einer bestimmten Region an einen bestimmten Kunden verkauft? Auch viele weitere Abfragen lassen sich auf diese Weise erstellen. Die Ergebnisse können dabei mehrere Dimensionen umfassen.

Das Schneeflockenschema stellt eine weitere Umsetzungsmöglichkeit dar. Dabei werden alle Tabellen normalisiert. Für die Dimensionen wird für jede Klassifikation eine Tabelle angelegt. Diese enthält neben der ID auch bestimmte Attribute zur Klassifikation. Die Kerndaten werden in einer Faktentabelle gespeichert. Die Faktentabelle verweist auf die jeweils niedrigere Klassifikation (vgl. Bauer und Günzel, 2013, S. 243 f.). Der Vorteil der normalisierten Tabellen liegt darin, dass sich dadurch Redundanzen vermeiden lassen und alle Informationen im Normalfall nur einmal vorhanden sind. Dies führt jedoch dazu, dass die Komplexität durch die Auslagerung schnell steigt.

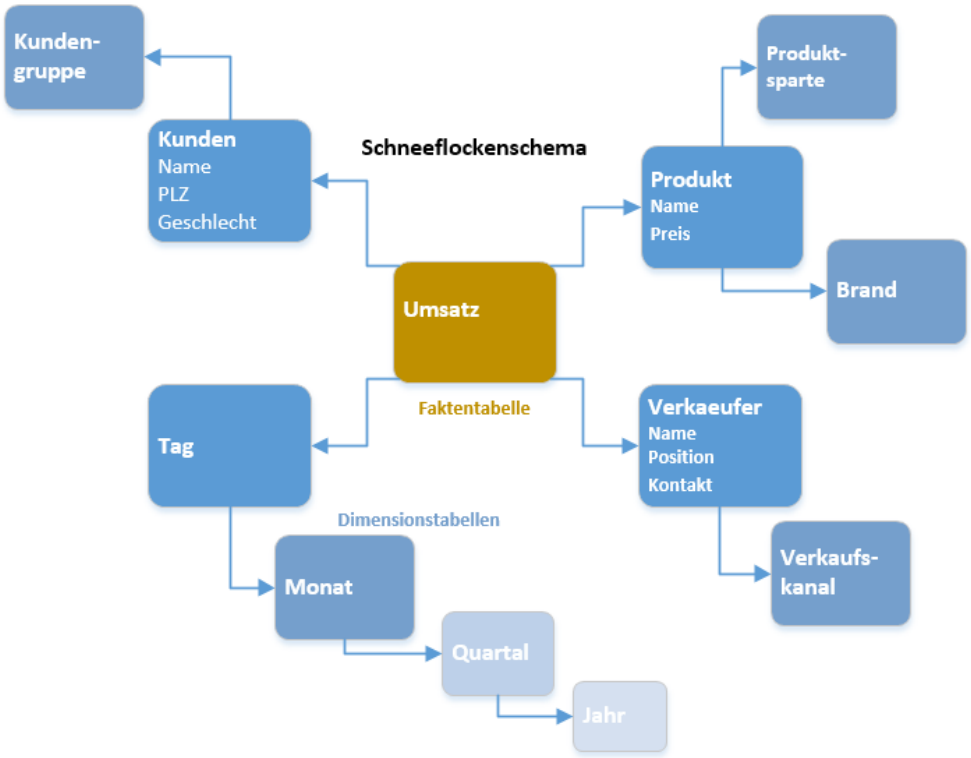


Abbildung 2.3.: Schneeflockenschema
Quelle: Modifiziert übernommen aus (Schnider u. a., 2016, S. 39)

Der Vorteil des Sternschemas, schnelle Abfragen zu ermöglichen, geht hier verloren. Abfragen im Schneeflockenschema sind auf Joins über die Dimensionstabellen angewiesen. Abbildung 2.3 zeigt eine vereinfachte Darstellung des Schneeflockenschemas. In der Praxis wird als Datenstruktur das Schneeflockenschema für Data Warehouse-Systeme genutzt. Das Sternschema wird aufgrund seiner schnelleren Abfragen für Data Marts benutzt.

2.4.2. Data Mart

Aufgrund der Menge an Daten, die in einem Data Warehouse gesammelt und gespeichert werden, ist eine schnelle Bearbeitung von Benutzerabfragen ab einem bestimmten Punkt nicht mehr zu gewährleisten. Daher ist das Ziel, die benötigen Daten die zur Verarbeitung notwendig sind, aus dem Data Warehouse zu extrahieren und somit einen kleineren Datenbestand zu halten. Die extrahierten Daten werden in Data Marts gespeichert, um Benutzern, Anwendungen oder Funktion zur Verfügung zu stehen (vgl. Chamoni und Gluchowski, 2016, S. 156 f.).

Ein Data Mart beinhaltet lediglich bestimmte Segmente aus dem Core Data Warehouse. Dies hat zur Folge, dass weniger Speicherplatz benötigt wird. Für die Erstellung von Data Marts kann es verschiedene Gründe geben. Neben dem bereits erwähnten Punkt, dass die Abfragen schnell und effizient durchgeführt werden sollen, gibt es noch weitere. In größeren Unternehmen ist nicht für alle Arbeitsbereiche der volle Zugriff auf den gesamten Datenbestand notwendig. Hier können Data Marts dazu verhelfen, den Fokus auf bestimmte Daten zu legen (vgl. Kemper u. a., 2010, S. 41).

Abbildung 2.4 veranschaulicht, wie sämtliche Daten, die einem Unternehmen zur Verfügung stehen, in einem Core Data Warehouse gespeichert werden. Diese Daten können von verschiedenen Bereichen abgefragt werden. So kann der Bereich Marketing auf Daten zugreifen, die für andere Bereiche nicht relevant sind. Eine weitere Einsatzmöglichkeit ergibt sich im Hinblick auf den Datenschutz. Nicht alle Bereiche in einem Unternehmen dürfen auf personenbezogene Daten zugreifen (vgl. Schnider u. a., 2016, S. 40). Die Umsetzung eines Data Marts hängt grundsätzlich von dem jeweiligen Anwendungsbereich ab. Generell ist eine hohe Anzahl separater Data Marts von Vorteil.

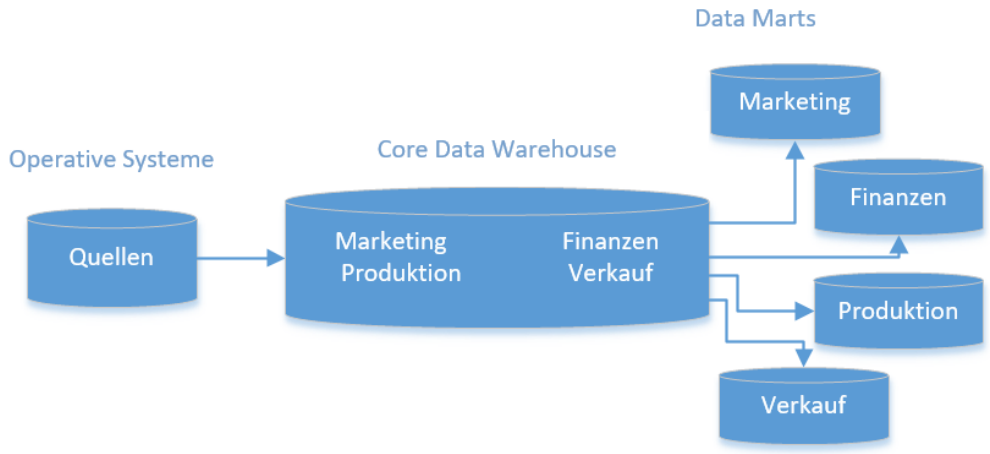


Abbildung 2.4.: Data Mart
Quelle: Modifiziert übernommen aus (Schnider u. a., 2016, S. 49)

Die Auswahl der Datenbanktechnologie ist auch vom jeweiligen Einsatzgebiet abhängig. Wie bei dem Core Data Warehouse kann auch bei Data Marts auf relationale Datenbanken zurückgegriffen werden. Es können jedoch auch multidimensionale Datenbank-Systeme (MOLAP)

genutzt werden. Bei diesen ist es von noch größerer Bedeutung, die Marts klein und die Komplexität gering zu halten.

2.4.3. Metadata-Repositorium

Ein Data Warehouse verfügt zur Speicherung der Metadaten über ein Metadaten-Repositorium. Der Ausdruck Metadaten bezeichnet im Allgemeinen bestimmte Daten, die für die Informationsverarbeitung gebraucht werden und Informationen enthalten, die für den Entwurf, die Konstruktion sowie die Benutzung eines Informationssystems notwendig sind (vgl. Eicker, 2011). So lassen sich Bücher in einer Bibliothek schneller über ihre Metadaten ausfindig machen – etwa über den Titel oder den Autor. Die Metadaten sind ein entscheidender Aspekt für ein erfolgreiches Data Warehouse-System, da diese Daten die Wartung und Administration erleichtern. Weiterhin lassen sich Daten mithilfe von Metadaten besser verstehen und verwalten. Dies ermöglicht dem Anwender eine schnelle und effiziente Suche nach Informationen.

Eine große Rolle spielt auch die Qualität der Daten. Hinsichtlich der Daten im Repositorium soll Anwendern oder Administratoren die Möglichkeit geboten werden, über Abfragen rasch Ergebnisse zu erzielen. Um diesen Ansprüchen gerecht zu werden, müssen die Daten vollständig sein. Weiterhin müssen die Daten korrekt sein. Nur so kann gewährleistet werden, dass bei Abfragen richtige Ergebnisse ausgegeben werden.

2.4.4. Data Warehouse-Management

Der Data Warehouse-Manager stellt eine weitere zentrale Komponente in einem Data Warehouse-System dar. Er steuert und überwacht sämtliche Prozesse, die in einem Data Warehouse-System anfallen. Dies beginnt bei dem Laden der Daten und reicht bis hin zur Auswertung der Daten in den Datenbanken. Weiterhin fallen der Aufbau, die Wartung und die Administration des Data Warehouse-Systems in den Aufgabenbereich des Managers. Die Beschaffung der Daten im Integrationsbereich kann vom Data Warehouse-Manager zu unterschiedlichen Zeiten durchgeführt werden kann (vgl. Müller und Lenz, 2013, S. 20). Sie kann in regelmäßigen Zeitintervallen erfolgen – zum Beispiel am Wochenende, wenn der Datenstrom abnimmt. Eine andere Möglichkeit besteht darin, zu warten, bis bestimmte Daten geändert wurden oder eine bestimmte Dateigröße erreicht ist. Schließlich kann sie auch explizit durchgeführt werden.

Der Data Warehouse-Manager agiert somit auf allen Ebenen des Data Warehouse-Systems. In der Datenerfassung werden die ETL-Prozesse überwacht. Weiterhin wird die Qualität der

Daten überprüft. Auf der Ebene der Datenhaltung überwacht der Data Warehouse-Manager die Speicherauslastung und administriert die Archivierung der Daten. Auf der obersten Ebene werden Abfragezeiten und administrative Funktionen überwacht, die zur Benutzerverwaltung dienen. Was die Sicherheit betrifft, werden Nutzerauthentifizierungen überwacht und Verschlüsselungen vorgenommen.

2.5. Zugriffskomponente

Nachdem alle Daten aufbereitet wurden und in das Core Data Warehouse geladen sind, können sie nun bereitgestellt werden. Die Daten können unmittelbar aus dem Core Data Warehouse stammen oder aus den Data Marts. Es kann nun auf verschiedenen Wegen mit den Daten gearbeitet werden. So können etwa Reports erstellt werden. Reports sind vordefiniert und stellen parametrisierte Berichte dar. Mit Reports können bestimmte Statistiken abgefragt werden – zum Beispiel der Quartals-Umsatz einer Firma. Reporting hat zum Ziel, die Frage zu beantworten, was in der Vergangenheit passiert ist. Es können aber auch neue Trends und Querverbindungen durch Data Mining gefunden werden. Eine weitere Möglichkeit zur Bereitstellung bietet OLAP.

2.5.1. Data Mining

Data Mining ist eine Methode zum automatischen Extrahieren von Zusammenhängen aus empirischen Daten. Mithilfe von Data Mining sind datengesteuerte Modellierungen und Explorationen möglich. Die Daten kommen dabei aus einer eigens dafür bereitgestellten Datenbank. Im Unterschied zu klassischen statistischen Verfahren werden im Bereich Data Mining große Datenmengen analysiert. Weiterhin soll Data Mining auch semi-automatisch funktionieren (vgl. Müller und Lenz, 2013, S. 75). Ziel dabei ist es, Informationen zu sammeln, die über die üblichen Kennzahlen des Controllings hinausgehen.

Die Ergebnisse lassen sich für unterschiedliche Zwecke nutzen. So kann der Zusammenhang von gekauften Produkten ermittelt werden. Es lässt sich jedoch ebenso gut feststellen, welche Kunden Artikel in ihrem Warenkorb haben und wie lange Kunden auf einer Website verweilen. Weiterhin lassen sich Aspekte ermitteln, die für die Kundentreue von Bedeutung sind. Durch Data Mining hat ein Unternehmen die Möglichkeit, aus vielen Daten Wissen zu gewinnen (vgl. Lackes, 2018). Da das Data Warehouse ähnlichen Zwecken dient, kann dieses gut durch Data Mining ergänzt werden. Auch die Tatsache, dass Unternehmen immer mehr Daten erzeugen,

spricht dafür, dass Data Mining-Methoden im Kontext von Data Warehousing zunehmend an Bedeutung gewinnen.

2.5.2. Online Analytical Processing

Online Analytical Processing (OLAP) ist ein Prozess, der zur Datenverarbeitung genutzt wird. Dabei können Nutzer gezielt bestimmte Daten extrahieren und sich diese aus verschiedenen Blickwinkeln anzeigen lassen. Die Daten werden dabei als OLAP-Würfel dargestellt. Vor diesem Hintergrund lassen sich dann die Analysen durchführen. Die Daten können unter anderem aus dem Core Data Warehouse stammen. Bei einem OLAP-Würfel handelt es sich um eine nicht relationale Datenstruktur. Es gibt auch relationale OLAP-Würfel (ROLAP), die auf der Basis von relationalen Datenbanken arbeiten, sowie solche, die auf Basis von multidimensionalen Datenbanken operieren. Diese letzteren werden als multidimensionale OLAP-Würfel (MOLAP) bezeichnet. ROLAP bietet sich für größere Datenmengen an. Für den Anwender ergeben sich daraus die Vorteile einer hohen Stabilität und Sicherheit. MOLAP erreicht eine gute Performance bei kleinen Datenmengen (vgl. Kemper u. a., 2010, S. 106 f.), da hier auf eine relationale Datenbank verzichtet wird. Ein OLAP-Würfel setzt sich aus Eigenschaften (Dimensionen) und Kennzahlen (Fakten) zusammen. Theoretisch kann ein solcher Würfel unbegrenzt viele Dimensionen umfassen (vgl. Gabriel u. a., 2009, S. 10 ff.). Ziel dabei ist es, einem Endanwender oder einer Führungskraft schnellen Zugriff auf komplexe Daten zu ermöglichen. Weiterhin sollen die Ergebnisse auch als Entscheidungshilfe dienen.

Abbildung 2.5 zeigt einen Würfel mit drei Dimensionen: Produkt, Region und Jahr. Die Achsen könnten jedoch auch andere Werte beinhalten. Aus den Ausprägungen der Dimensionen lassen sich Knotenpunkte herleiten, aus denen Kennzahlen abgelesen werden können. Auf diese Weise lassen sich Abfragen erstellen, die über drei Dimensionen gehen. In Abbildung 2.5 zeigt der rote Bereich den Umsatz aller verkauften DVD-Filme in Köln im Jahr 2015. OLAP-Würfel eignen sich gut zur Analyse von Daten und um Zusammenhänge besser erforschen zu können.

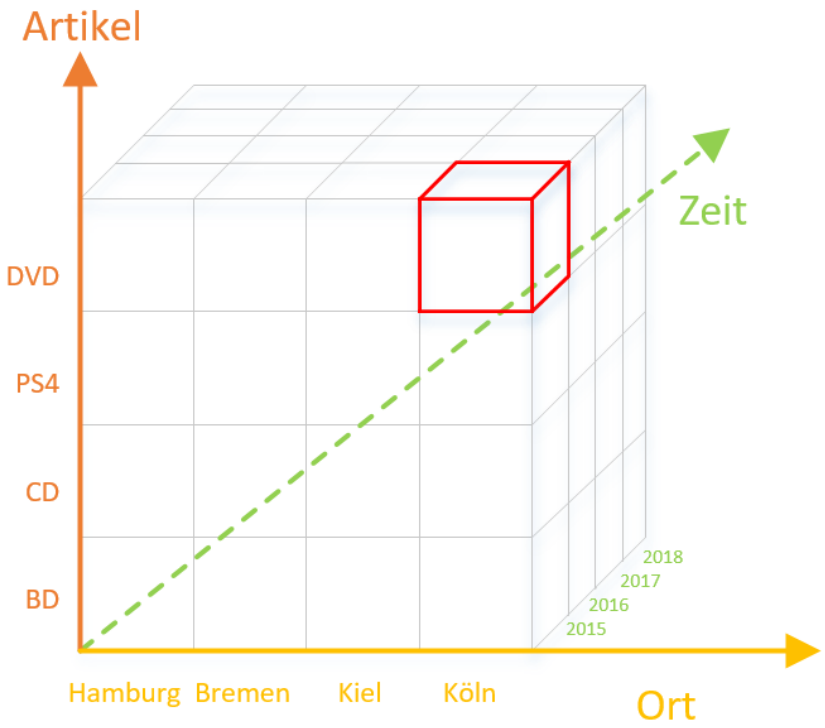


Abbildung 2.5.: OLAP-Würfel
Quelle: Modifiziert übernommen aus (Gabriel u. a., 2009, S. 58)

Um mit einen OLAP-Würfel umfangreich arbeiten zu können gibt es mehrere Methoden. Eine dieser Methoden ist das Slicing. Diese Methode ermöglicht es, einen Teil aus dem Würfel herauszuschneiden. So lassen sich die Informationen entsprechend einschränken und eine Konzentration auf bestimmte Aspekte erzeugen. Zum Beispiel könnte auf diese Weise die Frage beantwortet werden, wie der Umsatz für alle Produkte in der gesamten Region im Jahr 2015 ausfiel. Ein weiteres Ziel dabei ist, dass Regionalleiter, Abteilungsleiter und Analysten bei ihrer Arbeit nicht darauf angewiesen sind, sämtliche Daten durchzusehen, sondern vielmehr die Möglichkeit erhalten, diese in exakter Weise abzufragen und zu bearbeiten.

Eine andere Möglichkeit, mit einem OLAP-Würfel zu arbeiten, ist das Dicing. Hierbei wird ein Ausschnitt des Ganzen in einem kleineren Würfel dargestellt. Dicing wird für Ad-hoc-Abfragen genutzt und um Kennzahlen zu liefern – zum Beispiel für bestimmte Produkte in einer festgelegten Zeit und an einem bestimmten Ort (vgl. Bauer und Günzel, 2013, S. 124 f.). Weiterhin gibt es die Möglichkeit, den Würfel zu drehen, um so zu einer anderen Sicht auf die

Daten zu gelangen. Dieses Verfahren wird als Pivotierung oder Rotation bezeichnet.

Zur Arbeit in den Dimensionshierarchien gibt es die Operationen Roll-up und Drill-down. Bei der Operation Roll-up werden die Werte einer weiter unten liegenden Hierarchieebene mit denjenigen einer weiter oben liegenden Ebene zusammengeführt. In Abbildung 2.6 ist zu sehen, dass sich die Umsätze der Monate Januar, Februar und März zu dem 1. Quartal zusammenfassen lassen. Die Operation Drill-down bewirkt genau das Gegenteil. Hier werden die Werte wieder in ihre Bestandteile zerlegt (vgl. Kemper u. a., 2010, S. 102). Auf diese Weise ist es möglich, den Umsatz für das 1. Quartal wieder auf die drei Monate zu verteilen.

Während bei den Operationen Roll-up und Drill-down an den Hierarchieebenen gearbeitet wird, gibt es noch zwei weitere Operationen: Mit Drill-through und Drill-across können zusätzliche Informationen abgefragt werden. Drill-through dient dabei der Bereitstellung weiterer Details aus anderen Datenquellen. Dies passiert, ohne dass der Benutzer es bemerkt. Die Operation Drill-across ermöglicht noch tiefer reichende Abfragen. Sie basiert auf einer Verbindung von zwei Würfeln. Voraussetzung dafür ist, dass beide Würfel die gleichen Dimensionen aufweisen.

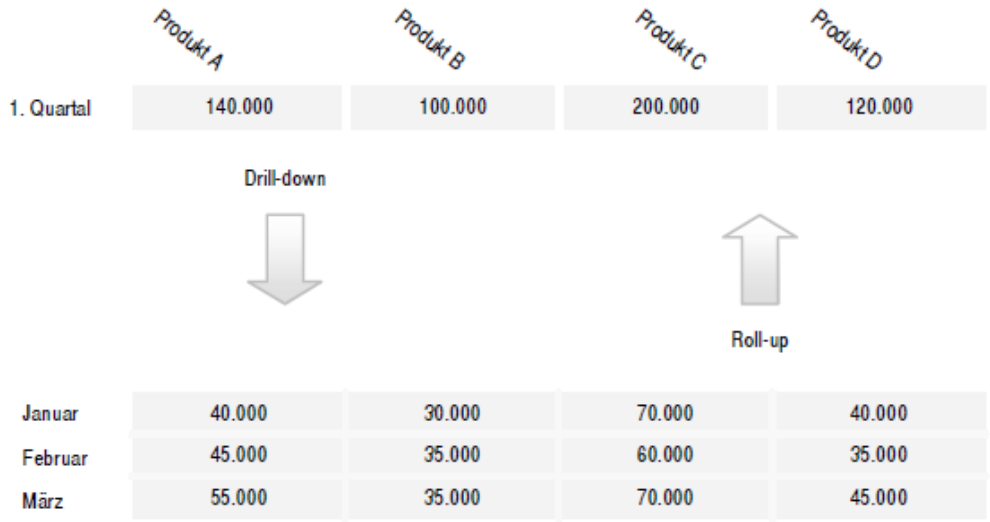


Abbildung 2.6.: Roll-up und Drill-down
Quelle: (Kemper u. a., 2010, S. 103)

2.5.3. Dashboard

Dashboards bieten dem Benutzer einen schnellen Überblick über alle nötigen Unternehmensinformationen. Weiterhin soll das Dashboard eine visualisierte Zusammenfassung zeigen, die sich möglichst auf eine Seite beschränken sollte. Auf diese Weise sind alle relevanten Daten schnell zur Einsicht bereit. Viele dieser Anwendungen sind über Browser oder mobile Endgeräte zugänglich.

Stephen Few definiert Dashboards wie folgt: „*A Dashboard is a visual display of the most important information needed to achieve one or more objectives, consolidated and arranged on a single screen so the information can be monitored at a glance* "(Few, 2013, S. 26).

Dashboards haben gegenüber rein textlichen Informationen den Vorteil, dass die Daten auf diese Weise schneller eingesehen werden können. So können auch Key Performance Indicators (KPI) in reduzierter Form enthalten sein. Durch farbliche Ampeln kann ein Anwender darauf aufmerksam gemacht werden, dass in bestimmten Bereichen eine Handlung erforderlich ist (vgl. Schnider u. a., 2016, S. 239 f.). Die Visualisierung bietet dabei einen weiteren Vorteil: Bilder und Grafiken vermitteln schneller Informationen an den Anwender als rein textbasierte Anzeigen.

Abbildung 2.7 zeigt die Vorteile eines Dashboards. Die einzelnen Bereiche sind farblich hervorgehoben, was verdeutlicht, welche Informationen gegeben sind. Das Erstellen und Anordnen der Elemente gestaltet sich unkompliziert. Auf diese Weise lassen sich Informationen schnell und ohne Schwierigkeiten zusammenführen und visualisieren. Meist werden viele Daten aus einem Data Warehouse benötigt, um die Informationen im Dashboard bereitstellen zu können. Alle relevanten Informationen sind auf der Seite zu sehen und lassen sich so überwachen und auswerten.

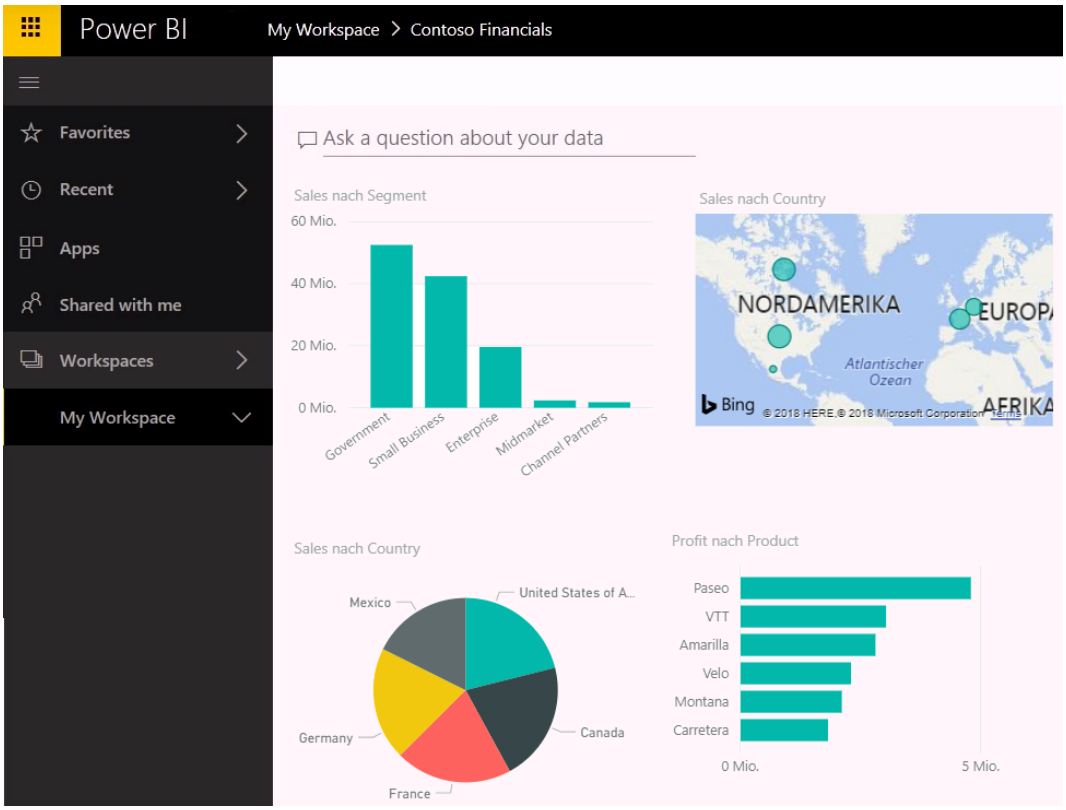


Abbildung 2.7.: Dashboard erstellt in Power BI

3. Moderne Data Warehouse-Systeme

In Kapitel 2 wurden die klassischen Data Warehouse-Systeme erläutert. Im Hinblick auf die wachsende Menge an Daten, die verarbeitet werden müssen, soll in diesem Kapitel erklärt werden, wie Data Warehouse-Systeme ersetzt oder ergänzt werden können. In Abschnitt 3.1 wird der Begriff Big Data erklärt. Zudem werden die Anforderungen von Big Data an Data Warehouse-Systeme erläutert und vertieft. Weiterhin wird erklärt, wie Big Data diese Systeme beeinflusst. In Abschnitt 3.2 werden dann die unterschiedlichen Technologien von Big Data erläutert. Die Merkmale von SQL und NoSQL werden verglichen, Hadoop sowie das Hadoop Distributed File System (HDFS) werden dargestellt. Im letzten Abschnitt 3.3 wird schließlich der Data Lake als Lösung für Big Data-Probleme aufgezeigt. Mit dieser Lösung gehen allerdings auch neue Probleme einher, auf die in diesem Zusammenhang näher eingegangen werden soll.

3.1. Big Data

Der Begriff ‚Big Data' nimmt zum einen auf die rasant wachsenden Datenmengen Bezug, die heutzutage durch Endgeräte, Internet und andere Quellen erzeugt werden. Zum anderen wird damit auf die neuen leistungsstarken IT-Lösungen und -Systeme angespielt. Da Unternehmen die Flut an Informationen effizient und vorteilhaft verarbeiten müssen, sind neue Technologien erforderlich. Stark unstrukturierte Daten – wie zum Beispiel aus dem Bereich von Social Media – machen einen großen Anteil davon aus. Dabei wollen Unternehmen einen Mehrwert aus den Daten ziehen und komplexe Abfragen erstellen, die wiederum einen wirtschaftlichen, gesellschaftlichen oder politischen Nutzen erbringen sollen. Durch die Sammlung von Daten versprechen sich Unternehmen Rückschlüsse auf das Kaufverhalten von Kunden, um auf dieser Grundlage Kundenprofile erzeugen zu können. Auch Behörden sammeln immer mehr und umfangreichere Daten, um diese auszuwerten. Solche Daten sollen zur Aufklärung oder Prävention dienen (vgl. Bendel, 2018). Die Kehrseite von Big Data ist, dass Daten von Menschen auch ohne deren Einverständnis gesammelt werden. Für den Datenschutz stellt Big Data insofern ein Problem dar, zumal Informationen, die über Menschen gesammelt und mit diesen in Verbindung gebracht werden, auch inkorrekt sein können. Eine Auswertung dieser fehlerhaften Daten kann zu falschen Schlüssen führen. Viele Menschen sind nicht damit

einverstanden, wenn personenbezogene Daten über sie gespeichert werden. Aus diesem Grund stehen viele Menschen Big Data kritisch gegenüber.

Big Data zeichnet sich durch drei Eigenschaften aus. Hierdurch wird deutlich, wie sich die Ansprüche an die Technik mit den Jahren verändert haben. Der Aspekt ‚Volume' nimmt dabei auf die ansteigende Größe der Daten Bezug. Weiterhin beschreibt ‚Variety' die Art und Vielfalt der erzeugten Daten. Die dritte Eigenschaft ist die ‚Velocity'. Hierbei geht es um die Geschwindigkeit, mit der Daten erzeugt werden (vgl. Klein u. a., 2013).

3.1.1. 3-V-Modell

Die Eigenschaften von Big Data wurden von der Gartner Group wie folgt definiert: „*Big data is high-volume, high-velocity and/or high-variety information assets that demand cost-effective, innovative forms of information processing that enable enhanced insight, decision making, and process automation.* " (Gartner, 2011).

Die Merkmale von Big Data und die damit einhergehenden Herausforderungen werden im Folgenden genauer ausgeführt.

Volume: Benutzer sozialer Netzwerke erstellen täglich neue Daten und speichern diese auf den Servern der Anbieter ab. Auf diese Weise werden große Mengen an Daten produziert, die über lange Zeit gespeichert werden. Die größten Datenmengen werden dabei von Anbietern wie Facebook oder Twitter erstellt. Doch auch andere Dienste wie E-Mails erzeugen Unmengen an Daten. Angesichts der benötigten Größe stoßen klassische Datenbanksysteme hier an ihre Grenzen. Zwar ist eine derartige Umsetzung weiterhin möglich, aber die Kosten für den Aufwand steigen dabei (vgl. Klein u. a., 2013). Zudem nimmt die Effizienz der Systeme bei solchen Datenmengen ab. Ein Beispiel für eine Plattform, die mit großen Datenmengen operiert, ist YouTube. Hier werden pro Minute über 300 Stunden an Videomaterial hochgeladen. Im Jahr 2016 wurden monatlich 6,2 Exabyte (6,2 Mrd. Gigabyte) an Daten durch mobile Geräte erzeugt (vgl. Firican, 2017).

Variety: Auch die Art der Daten hat sich mit der Zeit geändert, was einen der zentralen Aspekte von Big Data ausmacht. Klassische Datenbanksysteme haben Schwierigkeiten mit stark unterschiedlichen und nicht strukturierten Daten und können diese nicht effektiv speichern. Dabei ist zu unterscheiden zwischen strukturierten Daten (maschinengenerierten Daten), semistrukturierten Daten (E-Mails, bei denen der Kopf strukturiert und der Inhalt unstrukturiert ist)

und unstrukturierten Daten (Audio, Video und Bilder) (vgl. Meier, 2018, S. 6). Im Kontext von Big Data lassen sich auch unstrukturierte Daten durch NoSQL-Datenbanksysteme performant verarbeiten und abspeichern.

Velocity: Die Geschwindigkeit, mit der Daten in verschiedenen Bereichen erzeugt werden, ist drastisch angestiegen. Daher sollen Daten von einem System so schnell wie möglich – annähernd in Echtzeit – verarbeitet werden. Ziel dabei ist es, dass Unternehmen schnell auf eintretende Ereignisse reagieren können. Google liefert bei einer Anfrage durch einen Anwender sofort mehrere tausend Ergebnisse (vgl. Klein u. a., 2013), wobei allerdings nicht alle Ergebnisse relevant sind. Dabei kann der Einsatzbereich von einem Warenkorb in einem Online-Shop bis hin zur Betrugserkennung reichen. Damit diese Informationen schnell erfasst werden können, müssen die Daten rasch ausgewertet werden.

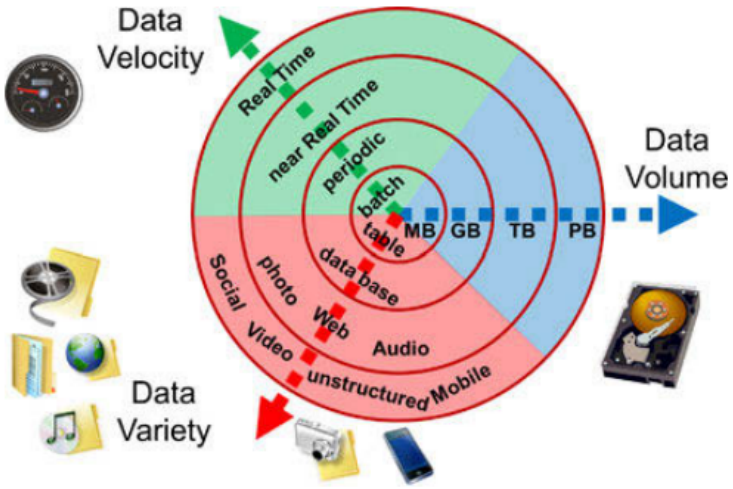


Abbildung 3.1.: 3-V-Modell
Quelle: (Klein u. a., 2013, S. 320)

In Abbildung 3.1 sind die drei Eigenschaften veranschaulicht, wobei die Entwicklung von innen nach außen voranschreitet. Die von Gartner definierten Eigenschaften wurden durch weitere ergänzt. Vier dieser neuen Eigenschaften sind die folgenden:

Value: Die Daten müssen für Unternehmen auch einen Wert besitzen und sollten bestenfalls den Unternehmenswert steigern (vgl. Fasel und Meier, 2018, S. 6). Nicht alle korrekten Daten

haben für Unternehmen einen Mehrwert. Unternehmen müssen sich selbst die Frage stellen, ob eine Investition in Personal und technische Infrastruktur einen Mehrwert zu generieren verspricht. Die Daten können zum besseren Verständnis der Kunden beitragen sowie zur Optimierung von Prozessen und der Verbesserung der Maschinen- oder Geschäftsleistung genutzt werden.

Veracity: Viele der Daten im Kontext von Big Data sind ungenau und müssen daher zunächst ausgewertet werden. Dabei stellt sich die Frage: Wie verlässlich sind die Daten? So wird auch die Qualität der Daten überprüft. Viele Daten beeinträchtigen die Qualität der Auswertung (vgl. Meier, 2018, S. 7).

Variability: Die Datenflussraten kommen in verschiedenen Variationen vor. Die Datengeschwindigkeit ist oft nicht konsistent, sondern hat periodische und aperiodische Spitzen. Dies macht es schwer, die Daten zu finden und zu säubern.

Visualization: Die Möglichkeiten und Techniken, Daten zu visualisieren, haben sich im Kontext von Big Data verändert und an Bedeutung gewonnen. Wie in Kapitel 2.5.3 beschrieben, haben Grafiken wie Balken- und Kreisdiagramme unterschiedliche Wirkungen und können die Daten auch im Bereich von Big Data auf verschiedene Weise repräsentieren (vgl. König u. a., 2016, S. 115 f.).

3.1.2. Kategorien von Big Data

Für die Wissenschaft, das Web oder die Unternehmen gibt es unterschiedliche Möglichkeiten, mit Daten umzugehen. Dabei gibt es aus Anwendungssicht verschiedene Aufgaben, die zu lösen sind. Unterschiedliche Datenquellen führen auch dazu, dass zur Problemlösung verschiedene Kategorien zur Verfügung stehen. Vier dieser Kategorien werden im Folgenden erläutert:

1. **Nachverfolgen und Auswerten** beschreibt den Vorgang, wie Daten für Big Data erfasst und bewertet werden können. Bei der Temperaturauswertung beispielsweise werden zunächst Daten gesammelt. Danach werden diese Daten ausgewertet, um Informationen über einen Tag, einen Monat oder ein Jahr zu erhalten. Auch für die Auswertung eines Standortes werden Daten – möglichst in Echtzeit – in ähnlicher Weise verarbeitet.

2. **Suchen und Identifizieren** stellt das Herausfiltern von Informationen aus einen großen Bereich an Daten dar. Dabei besteht die Schwierigkeit, dass die gegebenen Informationen

unvollständig oder ungenau sind. Die Suchfunktion von Google ist etwa darauf ausgelegt, schnell auf Benutzeranfragen zu reagieren. Dabei muss das Ergebnis trotz der gewaltigen Datenmenge zuverlässig der Absicht des Suchenden entsprechen.

3. **Analysieren** von Daten ist die am meisten genutzte Kategorie. In Bezug auf Big Data ist es für viele Unternehmen ein Anliegen, aus den vorhandenen Daten neues Wissen und Informationen abzuleiten. Zu diesem Zweck sind die Daten zu analysieren. Die häufigsten Einsatzbereiche bilden die künstliche Intelligenz sowie die Statistik.

4. **Vorhersagen und Planen** soll Unternehmen dabei helfen aus gesammelten Daten neue Informationen zu beziehen. Diese Informationen sollen es einem Unternehmen erlauben, Prognosen über die Zukunft anzustellen. Aus diesen Prognosen heraus lassen sich dann die Geschäftsprozesse optimieren (vgl. Freytag, 2014, S. 98 f.).

Die vier Kategorien lassen sich in unterschiedlichen Industriebereichen wiederfinden. Big Data wird nicht nur von IT-Unternehmen genutzt – auch im Bereich der Strom- und Wasserversorgung sowie im Gesundheitswesen. Aber auch für den Produktlebenszyklus sind diese Daten hilfreich. Bei kostspieligen Produkten wie Autos und Flugzeugen sind die gewonnenen Daten beispielsweise notwendig, um die Qualität der Produkte verbessern und auf diese Weise schneller auf Kundenwünsche eingehen zu können.

3.2. Big Data-Technologien

Zur Verarbeitung großer Datenmengen werden zahlreiche neue Technologien benötigt. Zu diesen gehören unter anderem die NoSQL-Datenbanken, die die klassischen SQL-Datenbanken bei der Verarbeitung von Audio-, Video- und Bild-Dateien ergänzen. Diese Datenbanken sind schemafrei und ermöglichen somit ein effizientes Speichern unstrukturierter Daten. Mit Hadoop wurde ein Framework entwickelt, das ein eigenes Dateisystem bereitstellt. HDFS ermöglicht eine Speicherung von großen Datenmengen auf einem Cluster von Computern. Zur parallelen Verarbeitung von Daten auf mehreren Computern wurde MapReduce als Programmiermodell entwickelt. Die neuen Technologien ermöglichen es, den in Abschnitt 3.1.1 erwähnten Anforderungen durch die drei Vs gerecht zu werden (vgl. Freytag, 2014, S. 100 f.).

3.2.1. NoSQL

Obwohl in vielen Unternehmen immer noch relationale Datenbanken effizient im Einsatz sind, stoßen diese bei großen verteilten Web- oder Big Data-Anwendungen an ihre Grenzen, sodass es sich als sinnvoll erweist, diese mit NoSQL-Technologien zu ergänzen. Durch diese Erweiterung können Dienste jederzeit und global angeboten werden. In diesem Zusammenhang ist von ‚NoSQL' bzw. ‚Not only SQL' die Rede. Besser ließe sich dieses System allerdings als ‚nicht relational' beschreiben, da NoSQL-Datenbanksysteme nicht mit relationalen Datenbankmodellen arbeiten (vgl. Meier, 2018, S. 10 f.). NoSQL-Datenbanksysteme wurden entwickelt, um eine Verarbeitung großer Datenmengen zu ermöglichen. Ab einem bestimmten Punkt können relationale Datenbanksysteme diese Herausforderung nicht mehr bewältigen. Vor allem bei großen Datenmengen, die nicht oft bearbeitet werden, erweisen sich NoSQL-Datenbanksysteme als effizienter. Beispielsweise sind solche Systeme für das Speichern von verschiedenen Informationen aus Twitter geeignet (vgl. Klein u. a., 2013, S. 321). Die Posts werden gespeichert und ausgewertet, aber nur selten bearbeitet.

Bei NoSQL-Datenbanksystemen werden die Daten nicht wie üblich in Tabellen gespeichert, sondern in Spalten, Graphen und Dokumenten. NoSQL-Datenbanksysteme lassen sich grundsätzlich in vier Kategorien einteilen.

1. **Key-Value**-Systeme bilden die unkomplizierteste Möglichkeit der Datenspeicherung. Dabei werden Daten in Schlüssel-Wert-Paaren abgespeichert.
2. **Column-Family**-Systeme stellen eine weitere Struktur dar. Die Daten werden hier spaltenweise und nicht wie üblich zeilenweise abgespeichert. Weiterhin werden die Spalten einer Tabelle zu Spaltenfamilien zusammengeführt.
3. **Graphdatenbanken** dienen zur Darstellung von netzwerkartigen Strukturen wie Karten oder sozialen Netzen. Die Daten werden in Kanten und Knoten abgespeichert. Die Graphen dienen zur Repräsentation der Daten.
4. **Document-Stores** speichern vollständige Dokumente ab. Die Dokumente können dabei in den Formaten JSON, XML und YAML vorliegen. Jedem Dokument wird dabei eine eigene ID zugewiesen (vgl. Fasel und Meier, 2018, S. 12).

Bei großen verteilten Datenbanksystemen hat es sich gezeigt, dass der Konsistenzanforderung nicht die höchste Priorität zukommt, da hier Verfügbarkeit und Ausfalltoleranz im Vordergrund

stehen. Bei webbasierten Anwendungen wird angestrebt immer verfügbar zu sein. Weiterhin sollte die Anwendung bei Ausfall eines Knotens weiter erreichbar sein. Solche Systeme haben geringe Anforderungen an die Konsistenz, sodass in diesem Zusammenhang auch von BASE die Rede ist (Basically Available, Soft State, Eventually Consistent). Bei BASE kommt der Verfügbarkeit die höchste Priorität zu. Bei solchen Systemen dürfen die einzelnen Knoten zwischenzeitlich verschiedene Versionen der Daten enthalten. Wenn alle Transaktionen abgeschlossen sind, befinden sich alle Daten wieder in einem konsistenten Zustand (vgl. Meier und Kaufmann, 2018, S. 148).

Im Jahr 2000 wurde die Vermutung von Eric Brewer (Universität Berkeley) aufgestellt, dass unmöglich alle drei Eigenschaften, Konsistenz (Consistency), Verfügbarkeit (Availability) und Ausfalltoleranz (Partition Tolerance), gleichzeitig in verteilten Rechnersystemen genutzt werden können (vgl. Brewer, 2000). Die drei Eigenschaften werden im Folgenden ausführlich erklärt:

Konsistenz (Consistency): Nach Abschluss der Transaktionen liegen sämtliche Daten in einem konsistenten Zustand vor. Wenn es also im Rahmen einer Transaktion zu bestimmten Änderungen kommt, wird der aktualisierte Datenstand sämtlichen lesenden Knoten zur Verfügung gestellt.

Verfügbarkeit (Availability): Die Anwendungen sind im Betrieb jederzeit verfügbar. Es können beständig Lese- und Schreib-Operationen ausgeführt werden. Diese Eigenschaft impliziert auch eine akzeptable Reaktionszeit, die für viele Shops von Bedeutung ist. Amazon und andere große Internethändler wären nicht so erfolgreich, wenn das Laden von Seiten mehrere Minuten in Anspruch nähme.

Ausfalltoleranz (Partition Tolerance): Wenn ein Knoten in einem stark verteilten System ausfällt, hat dies keinen Einfluss auf das Gesamtsystem. Weiterhin lassen sich jederzeit defekte Knoten austauschen und neue hinzufügen, ohne dass dadurch der laufende Betrieb gestört wird. Eine Verbindung des gesamten Systems ist jederzeit gegeben.

Im Jahr 2002 wurden die Theorie von Brewer durch Wissenschaftler des Massachusetts Institute of Technology (MIT) bewiesen und als CAP-Theorem bezeichnet (vgl. Gilbert und Lynch, 2002, S. 51 ff.).

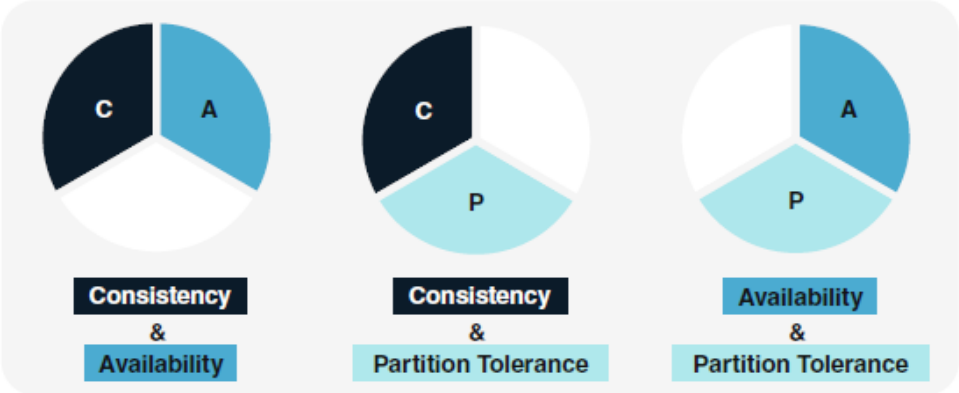


Abbildung 3.2.: Die drei Möglichkeiten des CAP-Theorems
Quelle: (Meier und Kaufmann, 2018, S. 149)

Das Theorem besagt, dass in einem großen verteilten Datenbanksystem jeweils nur zwei der drei Eigenschaften gleichzeitig gegeben sein können. Abbildung 3.2 zeigt die drei möglichen Kombinationen. Beispiele für die drei Möglichkeiten des CAP-Theorems werden im Folgenden erläutert:

1. Die Börse ist ein Beispiel für C+A. Alle Daten müssen hier durchgängig konsistent und die Verfügbarkeit muss jederzeit gewährleistet sein. Ohne diese Eigenschaften wäre ein Handel an den Börsen nicht effizient möglich.

2. Wenn ein Geldinstitut viele Geldautomaten im ganzen Land betreibt, handelt es sich dabei um C+P. Die Daten der einzelnen Automaten müssen durchgängig konsistent bleiben. Die Automaten müssen auch tolerant gegenüber Ausfällen im Netz sein. Sollte ein Teil des Netzes ausfallen, muss weiterhin gewährleistet sein, dass Kunden ihr Geld abheben können – auch wenn Kunden dann mit längeren Antwortzeiten rechnen müssen.

3. Das Domain Name System (DNS) ist ein Beispiel für A+P. Da das System die Hostnamen von Websites zu IP-Adressen auflöst, muss dieses ununterbrochen zur Verfügung stehen. Die Ausfalltoleranz ist hoch: Wenn ein Server ausfällt, beeinträchtigt dies einen Nutzer in keiner Weise. Allerdings fehlt es an Konsistenz. Geänderte Daten werden erst im Laufe der Zeit bei den Clients synchronisiert (vgl. Meier und Kaufmann, 2018, S. 149).

3.2.2. Hadoop

Hadoop ist ein Open Source Framework, das Daten erfasst, speichert, organisiert und analysiert. Die Daten können dabei eine unterschiedliche Struktur aufweisen. Sie werden auf einem Cluster von Rechnern gespeichert und dort verarbeitet. Die Architektur von Hadoop erlaubt eine schnelle und effiziente Verarbeitung großer Datenmengen und bietet eine hohe Skalierbarkeit. Das Framework verarbeitet die Daten batch-orientiert und bietet durch den Einsatz von vielen gewöhnlichen Computern ein gutes Preis-Performance-Verhältnis. Der Vorteil von Hadoop liegt im Kontext von Big Data vor allem darin, dass hierbei große Datenmengen auf kostengünstigen Computern sicher gespeichert werden können. Hadoop kann nicht als Datenbank angesehen werden. Es handelt sich dabei vielmehr um ein verteiltes Dateisystem, das die Daten im HDFS speichert. Mit Hadoop können Daten für längere Zeit gespeichert und ausgewertet werden. Für die parallele Verarbeitung wird MapReduce verwendet (vgl. Fasel und Meier, 2018, S. 145).

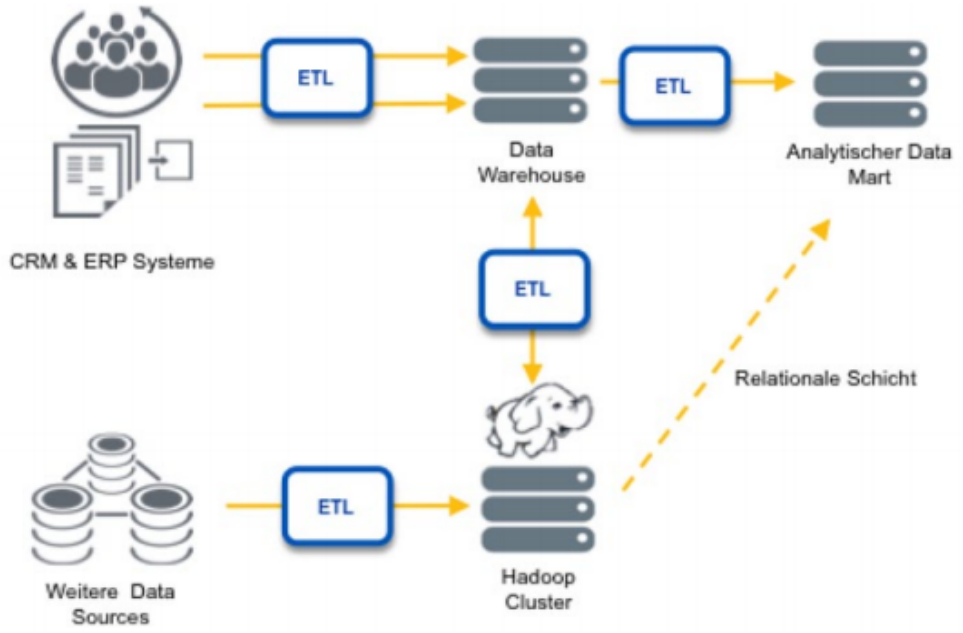


Abbildung 3.3.: Data Warehouse erweitert mit Hadoop im Kontext von Big Data
Quelle: (Müller, 2014, S. 455)

Abbildung 3.3 veranschaulicht beispielhaft, wie ein Data Warehouse-System durch Hadoop ergänzt werden kann. Die verarbeiteten Daten können dabei im Data Warehouse oder in den Data

Marts gespeichert werden. Die unverarbeiteten Rohdaten bleiben weiterhin im Hadoop-Cluster gespeichert. Der Vorteil einer Erweiterung mit Hadoop besteht darin, dass die Möglichkeit eines effizienten SQL-Zugriffs erhalten bleibt. Hadoop kann durch Hive erweitert werden und bietet dadurch eine SQL-ähnliche Abfragesprache an. Diese ist jedoch nicht so umfangreich wie SQL (vgl. Müller, 2014, S. 453 ff.).

Hadoop soll das Data Warehouse nicht ersetzen, sondern effizient ergänzen. Für die Analyse geringer Datenmengen ist Hadoop nicht geeignet. Weiterhin benötigen Unternehmen für die Anwendung im Hadoop-Umfeld neues Wissen und geschultes Personal. Diese neuen Ressourcen bringen nicht selten hohe Kosten mit sich und sorgen für ein neues Risiko im Unternehmen (vgl. Michalarias und Sümmchen, 2013).

3.2.3. MapReduce

MapReduce wurde als Modell in der Programmierung schon 2004 von Google vorgestellt (vgl. Dean und Ghemawat, 2004). Die Verarbeitung verteilt sich dabei auf verschiedene Rechner, sodass die Daten parallel berechnet werden. In Kapitel 3.2.2 wurde bereits erwähnt, dass Hadoop als verteiltes Dateisystem MapReduce zur Verarbeitung nutzt. In diesem Kapitel soll die Funktion von MapReduce vertiefend erläutert werden.

MapReduce bietet Vorteile bei großen verteilten Systemen, da die Operationen hierdurch auf viele verschiedene Computer verteilt werden können. Bei Operationen, für die ein einzelner Computer zu viel Zeit benötigen würde, hilft MapReduce die Daten bedeutend schneller zu verarbeiten. Große Cluster benötigen allerdings auch mehr Wartung und die Ausfälle im Verhältnis zu einem einzelnen Computer steigen gravierend an. Auch gestaltet sich die Fehlersuche in einem verteilten System schwieriger. MapReduce basiert auf einem unkomplizierten Modell: Die Map-Funktion speichert Daten als Key-Value-Paare. Die Dokumente können dabei den Key darstellen, das Value die einzelnen Wörter (vgl. Rahm u. a., 2015, S. 203 ff.).

Ein simples Beispiel für MapReduce ist das Zählen von Wörtern. Abbildung 3.4 veranschaulicht diese Funktionalität. Die Map-Funktion erhält als Eingabe eine Menge an Wörtern. Diese wird aufgeteilt und als Zwischenergebnis in Key-Value-Paaren gespeichert, wobei ein Wort der Key und dessen Wert 1 ist. Im nächsten Schritt werden die Daten sortiert, wobei doppelte Wörter zusammengeführt werden. Die Reduce-Funktion berechnet, wie oft ein Wort vorkommt, und addiert dabei gleiche Werte (Einsen). Als Ergebnis liefert MapReduce eine Summe der Werte,

also die genaue Häufigkeit der einzelnen Wörter in einem Dokument.

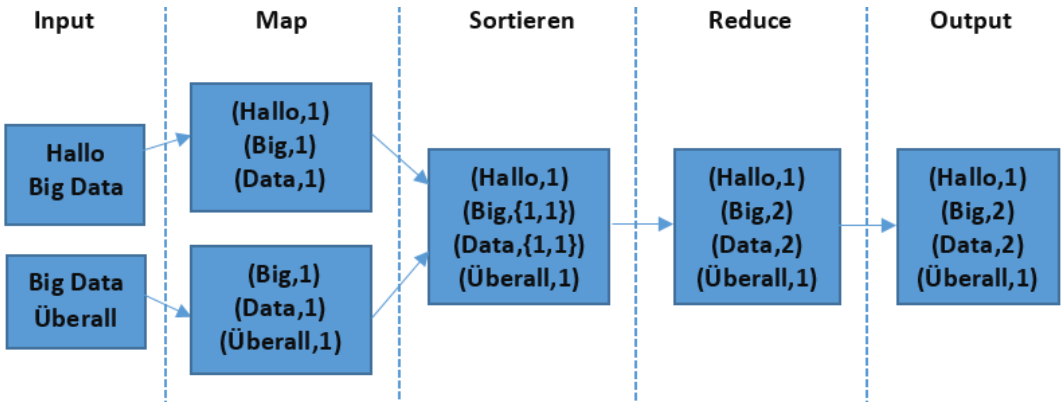


Abbildung 3.4.: Wörter zählen mit MapReduce
Quelle: Modifiziert übernommen aus (Rahm u. a., 2015, S. 205)

Doch stehen auch umfangreichere Funktionen zur Verfügung. Ein Beispiel dafür ist der Reverse Weblink Graph. Diese Funktion zeigt, wie eine Website mit anderen verlinkt ist. Dabei wird ein invertierter Graph erzeugt. Die Map-Funktion erhält eine Menge an Webseiten. Dabei werden Verlinkungen von dieser Webseite zu anderen gesucht. Nachdem die Daten zusammengefasst wurden, gibt die Reduce-Funktion eine Menge an URL's (Seiten) für jede einzelne Website aus (vgl. Schildgen u. a., 2013, S. 34 f.).

3.3. Data Lake

Bei dem Data Lake handelt es sich um eine Methode zur Speicherung von großen Datenmengen, die aus verschiedenen Quellen stammen und in ihrer Rohform abgespeichert werden. Die heterogenen Daten werden vor dem Ladeprozess weder gesäubert noch transformiert. Nachdem der Ladevorgang abgeschlossen wurde, liegen die Daten nun in einem einzigen System vor. In einem Data Lake werden neben strukturierten Daten auch semi-strukturierte (CSV, Logs, XML), unstrukturierte (E-Mails, Dokumente) und binäre Daten (Video, Audio, Bilder) gespeichert und verwaltet. Ziel dabei ist es, sämtliche Unternehmensdaten an einem Ort zu versammeln, um somit schnell auf den gesamten Datenbestand zugreifen zu können. Anwender sollen aus den Daten unmittelbar Visualisierungen, Reports und Analysen erstellen können. Dabei besteht die größte Herausforderung in der Verarbeitung der unstrukturierten

und heterogenen Daten (vgl. Isele und Arndt, 2016, S. 59 f.).

Die Tatsache, dass alle Daten in ihrer Rohform in einen Data Lake geladen werden und es vorher keine Bereinigung gibt, macht es umständlicher, die geforderte Datenqualität sicherzustellen. So kann es schnell dazu kommen, dass sich der Data Lake zu einem Datensumpf auswächst, der eine angemessene Auswertung erschwert (vgl. Pettey, 2015). Weiterhin erweist es sich als schwierig, die Zugriffsrechte für datenschutzrechtlich relevante Dokumente einzugrenzen, wenn alle Daten für sämtliche Benutzer an einem Ort zugänglich sind. Dies kann zu Problemen führen und datenschutzrechtlichen Anforderungen widersprechen. Daher muss stets geprüft werden, in welchem Umfang sich Kundendaten speichern lassen, um den Kundenbedürfnissen besser gerecht werden zu können, ohne dadurch in Widerspruch mit dem Datenschutz zu treten (vgl. Gimpel u. a., 2018, S. 94).

Damit das Prinzip des Data Lakes für Unternehmen effizient funktioniert und keinen Datensumpf zur Folge hat, in dem keine Daten mehr gefunden werden, müssen die gesammelten Daten einen unternehmerischen Mehrwert für die Zukunft aufweisen. Für Analysten ist es schwer, aus der Menge an Daten Informationen zu gewinnen. Dies gilt vor allem dann, wenn keine Metadaten vorliegen. Ohne diese ist es für Analysten kaum möglich, die Daten zuzuordnen. Weiterhin ist es nahezu unmöglich, aus den Daten Beziehungen abzuleiten, da alle Daten an einem Ort gespeichert werden und keinen sichtbaren Zusammenhang besitzen. Ein erster Schritt, um einen besseren Überblick zu gewinnen, sind Metadaten. Auf diese Weise können zum Beispiel Besuche, Aufrufe und Käufe auf einer Website mit IP-Adressen und Standorten verbunden werden. Metadaten können in einem Data Lake also als Roadmap zu den relevanten Daten verstanden werden (vgl. Inmon, 2016).

Abbildung 3.5 zeigt den beispielhaften Aufbau eines Data Lakes. Wie bei Data Warehouse-Systemen kommen die Daten auch hier aus vielen unterschiedlichen Quellen. Es handelt sich dabei jedoch überwiegend um Daten aus dem Bereich Big Data – also Daten wie E-Mails, Dokumente, Videos und Bilder. Den Kern eines solchen Systems bildet ein Repositorium, das sämtliche Daten beinhaltet. Ab diesem Punkt können die Daten dann aus dem Data Lake entnommen und transformiert werden. Im vorliegenden Beispiel werden die transformierten Daten außerhalb des Data Lake-Systems in Data Marts gespeichert (vgl. Benghiat, 2017). Anwender müssen sich darüber im Klaren sein, welche Daten sie aus dem Data Lake benötigen, um dann mit diesen ausgewählten Daten weiterarbeiten zu können.

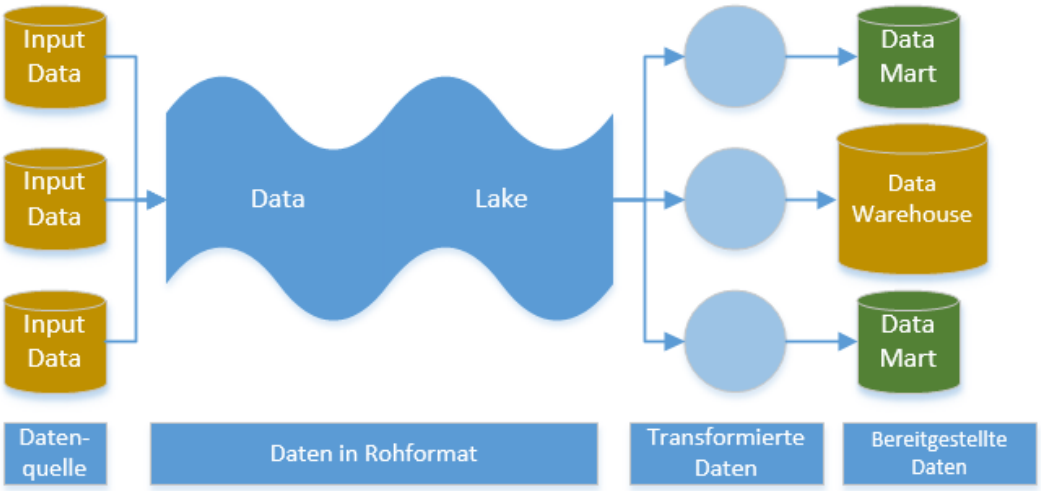


Abbildung 3.5.: Data Lake
Quelle: Modifiziert übernommen aus (Benghiat, 2017)

Für die Bearbeitung ist es teilweise nötig, Informationen annähernd in Echtzeit zu verarbeiten. Eine dahingehende Notwendigkeit kann sich etwa bei solchen Daten ergeben, die von Sensoren, sozialen Medien oder anderen dynamischen Quellen erzeugt werden. Analysten wollen mit eingehenden Daten sofort arbeiten und die Ergebnisse so schnell wie möglich auswerten. Durch die Zwischenspeicherung der Daten sowie die Verarbeitung im Batchbetrieb kommt es zu starken Verzögerungen, die eine Verarbeitung in Echtzeit unmöglich machen. Dabei sollen die generierten Daten allerdings auch dauerhaft gespeichert werden. So können diese Daten auch Jahre später für Analysen genutzt werden (vgl. Mathis, 2017, S. 292 f.).

3.3.1. Extract-Load-Transform-Prozess (ELT-Prozess)

Bei Data Lake-Systemen werden die Daten zuerst in das System geladen, bevor sie dann transformiert werden. Auf diese Weise lässt sich sicherstellen, dass auch tatsächlich alle Informationen erfasst werden. Da die Daten in Rohform vorliegen, müssen sie vor der Benutzung erst aufbereitet werden. Dabei findet eine Verschiebung der Reihenfolge statt. Aus dem ETL-Prozess – siehe Abschnitt 2.3.2 – wird ein ELT-Prozess. Da die Daten vor dem Laden nicht gesäubert und umgewandelt werden müssen, lässt sich jederzeit eine neue Auswertung ergänzen. Nachdem alle Daten in einen Data Lake geladen wurden, können diese für den Einsatz transformiert werden. Dabei ist zu entscheiden, ob die Ergebnisse in einem Data Mart oder im Data Lake selbst gespeichert werden.

Abbildung 3.6 veranschaulicht den Ablauf des ETL-Prozesses. Nachdem bestimmte Daten aus dem Data Lake gesammelt und transformiert wurden, lassen sich aus diesen weitere Informationen gewinnen. Diese lassen sich in Dashboards, Reports und anderen Anwendungen darstellen. Die Ergebnisse der Transformation können wahlweise wieder im Data Lake gespeichert werden. Zudem ist es möglich, die Daten extern in Data Marts zu speichern, wie dies in Abbildung 3.5 veranschaulicht wird. Bei solchen Systemen liegt das Gewicht vor allem auf der Velocity (Geschwindigkeit), einem der drei großen Vs. Auf diese Weise können Abweichungen im System schneller erkannt und Betrugsversuche verhindert werden. Im gewählten Beispiel werden die verarbeiteten Daten nicht in einem externen Data Mart gespeichert, sondern in dem Data Lake selbst (vgl. Soutier, 2015).

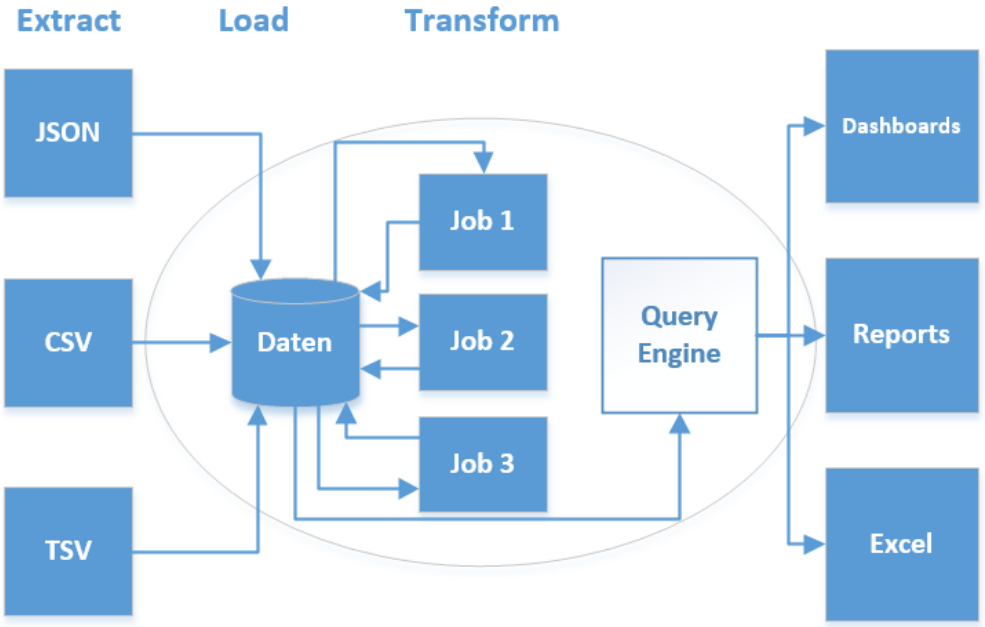


Abbildung 3.6.: ELT-Prozess
Quelle: Modifiziert übernommen aus (vgl. Soutier, 2015)

Welcher Prozess am besten geeignet ist, hängt von dem jeweiligen Einsatzzweck ab. Im Folgenden werden einige Eigenschaften erläutert, im Hinblick auf die Data Warehouse- und Data Lake-Systeme dann verglichen werden sollen:

1. **Laufzeit**: Das Laden der Daten nimmt bei einem Data Warehouse mehr Zeit in Anspruch, da die Daten hier erst über die Staging Area verarbeitet werden. Bei einem Data Lake werden die Daten dagegen direkt geladen. Der Zugriff auf neue Daten ist somit schneller.

Für die Transformation großer Daten benötigt ein Data Warehouse länger. In einem Data Lake sind alle Daten unmittelbar vorhanden. Eine Transformation fällt ausschließlich für die jeweils benötigten Daten an.

2. **Komplexität der Implementation**: Data Lake-Systeme setzen ein umfangreiches Wissen über neue Technologien wie auch über den Aufbau des Haupt-Repositoriums voraus, in dem alle Daten gespeichert werden. Data Warehouse-Systeme basieren auf etablierten Technologien, die es schon lange gibt. Hierdurch erweist sich die Implementierung als unkomplizierter als dies bei den neuen Technologien der Fall ist, die für Data Lake Systeme benötigt werden.

3. **Data Warehouse-Unterstützung**: ETL-Prozesse speichern strukturierte Daten in relationalen Datenbanken. ELT-Prozesse können verwendet werden, um das System zu erweitern. Es bietet eine skalierbare Infrastruktur, die neben strukturierten auch unstrukturierte Daten unterstützt.

4. **Data Lake-Unterstützung**: ELT-Prozesse ermöglichen die Benutzung von unstrukturierten Daten sowie eine Speicherung und Verwaltung großer Datenmengen, die im Kontext von Big Data erzeugt wurden. Für Data Lake-Systeme sind ETL-Prozesse nicht geeignet. Diese würden dem Grundsatz widersprechen, alle Daten in ihrer Rohform zu speichern.

5. **Benutzbarkeit**: Im Vergleich mit Data Warehouse-Systemen bieten Data Lake-Systeme mehr Agilität und Flexibilität. Da alle Daten in ihrer Rohform gespeichert werden, stehen für Abfragen die Originaldaten zur Verfügung.

Der ELT-Prozess wird für Unternehmen, die viele unstrukturierte Daten erzeugen und diese unbearbeitet speichern wollen, immer interessanter. Auf diese Weise können sämtliche Daten, die im Kontext von Big Data erzeugt werden, erst einmal gespeichert werden. Danach können Anwender die benötigten Daten auswählen, um dann mit diesen weiterzuarbeiten (vgl. Avinoam, 2018).

4. Realisierung von Data Warehouse- und Data Lake-Systemen

In Kapitel 2 wurden die klassischen Data Warehouse-Systeme vorgestellt. Dabei wurden die einzelnen Komponenten und der Aufbau erläutert. Diese Systeme arbeiten mit SQL-Datenbanken und müssen im Kontext von Big Data und den neuen Herausforderungen durch neue Technologien ergänzt werden. In Kapitel 3 wurden moderne Data Warehouse-Systeme dargestellt. Dabei wurde zum einen auf Data Warehouse-Systeme eingegangen, die durch neue Technologien erweitert wurden. Zum anderen wurde der Data Lake als Ergänzung vorgestellt. Beide Systeme sind dazu in der Lage, große Datenmengen zu speichern und zu verarbeiten. Dabei gibt es jedoch Unterschiede im Umgang mit dem Daten – vor allem bei der Reihenfolge, in der diese jeweils aufbereitet werden. In diesem Kapitel soll der Unterschied zwischen beiden Systemen herausgearbeitet werden, wobei der Schwerpunkt auf den Anforderungen durch Big Data liegt. Klassische und moderne Data Warehouse-Systeme können auf unterschiedliche Weise große Datenmengen speichern und verarbeiten. In Abschnitt 4.1 werden beide Systeme direkt gegenübergestellt, um so die gravierendsten Unterschiede zu erfassen. Weiterhin wird in Abschnitt 4.2 auf den Umfang der Realisierung eingegangen. Bei der Realisierung werden Anwendungen von Google (siehe Abschnitt 4.3), Microsoft (siehe Abschnitt 4.4) und Amazon (siehe Abschnitt 4.5) vorgestellt. Am Ende wird in Abschnitt 4.6 eine Bewertung dieser Systeme vorgenommen.

4.1. Kernunterschiede beider Systeme

In den Kapiteln 2 und 3 wurden die beiden Systeme Data Warehouse und Data Lake vorgestellt. In diesem Abschnitt sollen nun die Unterschiede beider Systeme aufgezählt und diese an einigen Beispielen erläutert werden. Dabei werden zentrale Aspekte, die für beide Systeme relevant sind, miteinander verglichen. In der folgenden Tabelle 4.1 werden die Kernunterschiede gegenübergestellt.

Data Warehouse	**vs.**	**Data Lake**
strukturierte und verarbeitete Daten	Daten	strukturierte, semi-strukturierte und unstrukturierte Daten
Schema on Read	Datenaufbereitung	Schema on Write
Bereinigung der Daten vorher notwendig	Speicherung	Daten werden in Rohform gespeichert
weniger Agil, festgelegte Konfigurationen	Schnelligkeit	hohe Agilität, Konfiguration wenn benötigt
ausgereift	Sicherheit	im Reifeprozess
Business Anwender	Benutzer	Data Scientists und andere Anwender

Tabelle 4.1.: Gegenüberstellung Data Warehouse und Data Lake

Diese Unterschiede werden im Folgenden weiter erläutert:

1. **Daten:** In einem Data Warehouse werden nur solche Daten gespeichert, die vorher gesäubert und transformiert wurden – Daten, die für das Core Data Warehouse modelliert und strukturiert wurden. Dann lassen sich auf diesen zum Beispiel Daten analysieren oder Reports und Visualisierungen erstellen. In einen Data Lake hingegen werden alle Daten in ihrer Rohform geladen. Vor dem Ladevorgang wird demnach keine Aufbereitung vorgenommen. Wie in einem Data Warehouse werden auch hier insbesondere strukturierte Daten geladen, aber auch semistrukturierte und unstrukturierte Daten.

2. **Datenaufbereitung:** Bevor die Daten in das Data Warehouse geladen werden können, müssen sie bearbeitet und strukturiert werden. Bevor die Daten geladen werden, werden sie in ein bestimmtes Schema überführt. Dieser Vorgang wird als ‚Schema-on-write' bezeichnet. In einen Data Lake werden die Daten in ihrer Rohform geladen. Dabei spielt es keine Rolle, aus welcher Quelle die Daten stammen oder welche Struktur diese aufweisen. Erst wenn die Daten gebraucht und benutzt werden, müssen diese aufbereitet werden. Dieser Vorgang wird als ‚Schema-on-read' bezeichnet.

3. **Speicherung:** Ein Data Lake benötigt für die Verarbeitung von Daten im Bereich von Big Data neue Technologien. Eine dieser Technologien ist Hadoop (siehe Abschnitt 3.2.2). Die Kosten für die Datenspeicherung fallen hier im Vergleich zum Data Warehouse niedrig aus. Dies liegt zum einen daran, dass Hadoop Open Source ist, sodass Lizenz und Support kostenlos sind. Zum anderen ist Hadoop für die Installation auf kostengünstiger Hardware konzipiert.

4. **Schnelligkeit:** Die Aufbereitung der Daten in einem Data Warehouse nimmt viel Zeit in Anspruch. Benutzer haben in dieser Zeit nicht die Möglichkeit, auf diese Daten zuzugreifen. Sie müssen warten, bis die Daten aufbereitet und zur Verfügung gestellt wurden. Im Gegensatz dazu stehen die Daten in einem Data Lake zur sofortigen Verfügung bereit – annähernd in Echtzeit. Der Nachteil dabei ist die Masse an unstrukturierten Daten. Nicht jeder Benutzer weiß mit den unstrukturierten Daten etwas anzufangen. Ein Vorteil liegt darin, dass Data Scientists ihre Abfragen auf die jeweils benötigten Daten beschränken können, jedoch alle Daten weiterhin zur Verfügung stehen. So ist es unproblematischer, diese zu konfigurieren und zu rekonfigurieren.

5. **Sicherheit:** Data Warehouse-Systeme gibt es schon lange. Dementsprechend ist die Sicherheit der Daten in diesen Systemen viel weiter ausgereift als in Data Lake-Systemen. Die Technologien im Bereich von Big Data sind noch vergleichsweise jung und daher wenig ausgereift. In diesem Sinne befindet sich die Sicherheit noch in der Prüfung und wird stetig verbessert. Derzeit werden erhebliche Anstrengungen im Bereich der Sicherheit unternommen.

6. **Benutzer:** Data Warehouse-Systeme sind auf Reports und Metriken ausgelegt. Unternehmen sollen dadurch Auswertungsmöglichkeiten erhalten. Zudem sollen auf diese Weise Informationen generiert werden. Ein Data Warehouse speichert nur solche Daten, die darauf ausgelegt sind. Data Lake-Systeme sind am besten für Data Scientists geeignet, da die Daten hier nicht limitiert sind. Anwender können mit ungefilterten Daten arbeiten und Analysen durchführen. Während der Data Scientist in einem Data Lake in die Tiefe steigt, kann der Business-Benutzer an der Oberfläche arbeiten. Ein Data Lake ist für beide gleichermaßen von Nutzen (vgl. Knight, 2017).

Die Hauptaufgabe beider Systeme besteht darin, Daten abzuspeichern, um Informationen aus diesen zu gewinnen. Sie können beide von Unternehmen genutzt werden. Bei der Auswahl sollten Unternehmen bedenken, welches Ziel durch die Auswertung der Daten erreicht werden soll. Ein Data Lake kann also weder als Data Warehouse 2.0 gesehen werden noch als Ersatz für ein Data Warehouse. Die beiden Systeme sind für unterschiedliche Verwendungszwecke optimiert. Ziel sollte es dabei sein, jedes der Systeme zu dem Zweck zu nutzen, für den es entworfen wurde (vgl. Dull, 2015). Für einen Data Lake sprechen in der Zukunft die immer größeren und schneller erzeugten Daten. Diese Daten sollen weiterhin so schnell wie möglich verarbeitet werden. Auch im Bereich ‚Machine Learning' werden beständig neue Daten erzeugt. Diese Daten werden auch als Quellen berücksichtigt und müssen in die Analyse miteinfließen (vgl. Gadatsch und Landrock, 2017, S. 5).

4.2. Realisierungsumfang

Die Umsetzung eines Data Lake-Systems gestaltet sich umfangreich und anspruchsvoll. Es gibt verschiedene Wege, ein solches System aufzubauen. Zum einen können alle Komponenten selbst aufgesetzt werden. Zu diesem Zweck wird allerdings ein umfangreiches Know-how in vielen verschiedenen Bereichen sowie eine gut ausgebaute Infrastruktur für Big Data benötigt. Zum anderen bieten Internetanbieter eben diese Infrastruktur an. Firmen wie Google, Amazon und Microsoft bieten Komponenten oder Komplettlösungen für kleinere und mittlere Unternehmen an. Die Vorteile liegen unter anderem in den Kosten, der bereitgestellten Infrastruktur und den vorhandenen Anwendungen. Bei diesem Vorgehen benötigt ein Unternehmen keine vollständig selbst erstellte Infrastruktur. Weiterhin bieten diese Firmen auch technische Unterstützung an. Im Rahmen der vorliegenden Arbeit werden Teilkomponenten von Cloud-Anbietern praktisch umgesetzt und erläutert. Es wurden Anwendungen ausgewählt, die den Ablauf von Data Warehouse- oder Data Lake-Systeme darstellen. In dieser Arbeit werden drei Anbieter miteinander verglichen. Somit werden nur Teile dieser Systeme umgesetzt. Durch diese Umsetzungen wird ein Einblick in den Ablauf der Speicherung, Verarbeitung und Bereitstellung solcher Systeme gegeben. Für die Realisierung werden Google Cloud, Microsoft Azure und Amazon Web Services untersucht. Nach Abschluss der Untersuchung werden die Ergebnisse verglichen und bewertet.

4.2.1. Datenquelle

Die zur Verarbeitung ausgewählten Daten stammen von der Federal Election Commission und sind damit frei zugänglich (Hunter, 2018). Die Daten beinhalten mehrere Spalten und mehrere Millionen Zeilen. Es wurde eine Quelle benutzt, die typisch für Big Data ist (siehe Abschnitt 3.1). Solche Daten beinhalten viele leere Zellen, die vor der Weiterverarbeitung gesäubert werden müssen. Weiterhin existieren viele Spalten, die nicht relevant sind. Aus diesem Grund müssen alle Spalten selektiert werden, die für die Verarbeitung von Relevanz sind. Nicht alle Spalten aus Quellen sind für ein Unternehmen interessant und enthalten neue Informationen. Im ersten Schritt werden die Daten importiert. Hier wurden zwei Daten für die Bearbeitung ausgewählt. Im ersten Dokument sind alle Kandidaten und Parteien aufgeführt, die für die Wahlen 2016 registriert waren. Im zweiten Dokument sind alle Einzelspenden für politische Kampagnen aus dem Jahr 2016 festgehalten. Bei der Bearbeitung werden die beiden ausgewählten Quellen immer in ihrer Rohform verwendet, um eine Vergleichbarkeit der Ergebnisse zu gewährleisten.

Viele Aspekte wiederholen sich bei den verschiedenen Umsetzungen. Daher wird nur im Rahmen der ersten Umsetzung detailliert auf alle Eigenschaften eingegangen. Bei den folgenden Umsetzungen werden nur die Unterschiede hervorgehoben, sodass es zu weniger Wiederholungen kommt.

4.3. Experimentelle Untersuchung: Google Cloud

Die erste Untersuchung beinhaltet drei Komponenten, die von Google angeboten werden. Bei dieser Umsetzung werden die Daten, die in Kapitel 4.2.1 vorgestellt wurden, bearbeitet und visualisiert. Google biete viele weitere Komponenten an. Zur Speicherung wurde die Cloud Storage genutzt. Diese speichert vollständige Dokumente ab und fällt in die Kategorie der Document-Stores von NoSQL-Datenbanken (siehe Abschnitt 3.2.1). Zur Bereinigung wird die Anwendung Dataprep verwendet, für die Bearbeitung und Speicherung BigQuery sowie schließlich für die Visualisierung Data Studio.

4.3.1. Bereinigung der Daten mit Cloud Dataprep

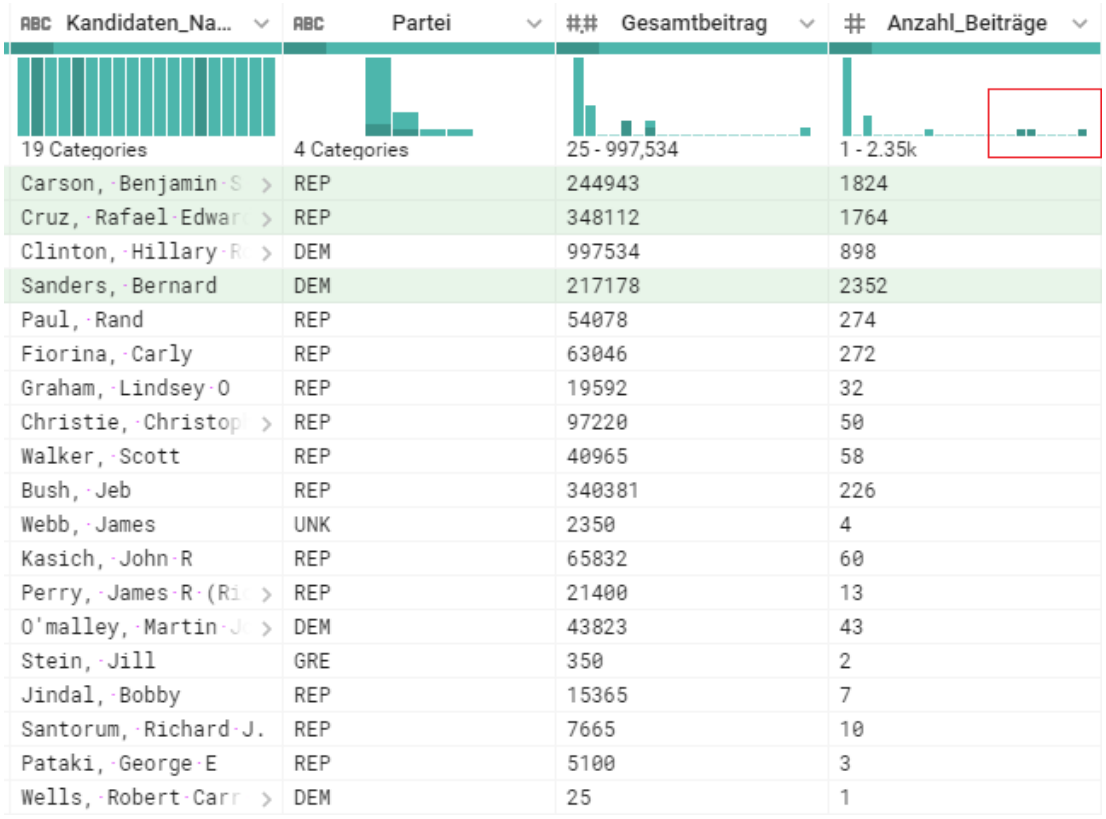
Durch die Anwendung Cloud Dataprep werden Daten bereinigt, überprüft und für die weitere Bearbeitung vorbereitet. Die Daten können strukturiert oder unstrukturiert sein. Für die Aufbereitung werden fehlende Datensätze, Duplikate und Fehler erkannt und bereinigt. Im ersten Schritt wird ein Recipe erzeugt. Dieser speichert relevante Daten und gruppiert diese. Danach werden die Ergebnisse nach der Transformationslogik zusammengeführt. Zuerst wird die Kandidatenliste bereinigt und transformiert. Dabei wird das Format einer Spalte so angepasst, dass die Werte als String erkannt werden. Weiterhin werden unnötige Daten angepasst oder entfernt. Die Spalten mit der Straße und der Hausnummer werden zusammengeführt. Im letzten Schritt werden dann die gewünschten Daten extrahiert.

Im zweiten Recipe werden die Gesamtbeiträge des Präsidentschaftskandidaten angezeigt. Dabei wird zunächst ein Inner Join durchgeführt, der die Kampagnenbeiträge 2016 mit der gefilterten Kandidatenliste verbindet. Danach werden die Zahlen für jeden einzelnen Kandidaten summiert und zusätzlich ein Durchschnitt gebildet. Für jede dieser Aggregationen wird eine neue Spalte erstellt. Danach werden alle benötigten Spalten in sinnvoller Weise benannt. Weiterhin werden alle Kommazahlen entfernt, was die Lesbarkeit verbessert. Abbildung 4.1 zeigt den fertigen Flow mit zwei Recipes und wie diese miteinander verbunden sind.

Abbildung 4.1.: Data Flow in Dataprep
Quelle: Eigene Darstellung in Cloud Dataprep

Aus dem Ergebnis lässt sich die Analyse der Spende der Präsidentschaftskampagne 2016 in den USA ablesen. Es wird die Gesamtzahl der durchschnittlichen USD-Spenden pro Kandidat berechnet. In Abbildung 4.2 sind alle Daten nach der Bereinigung und Transformierung zu sehen. Die Daten können auch in Dataprep gefiltert werden. Ein Beispiel dafür ist die Anzahl der Beiträge bei den obersten drei Kandidaten. In Abbildung 4.2 ist die Auswahl rot markiert. In Grün ist das Ergebnis zu sehen.

Abbildung 4.2 zeigt eine kleine Tabelle, die nur zu Demonstrationszwecken dienen soll. Auf diese Weise soll veranschaulicht werden, welche Filter-Möglichkeiten Dataprep bietet. Für die weiteren Schritte werden Quellen benutzt, die weniger stark aufbereitet und gesäubert wurden, sodass diese Daten immer noch mehrere Millionen Zeilen umfassen. Dafür wird auf die beiden in 4.2.1 eingeführten Dokumente zurückgegriffen. Im ersten Schritt werden beide Dokumente miteinander verbunden. Danach werden 29 Spalten entfernt, die für die weitere Bearbeitung irrelevant sind. Für alle übrigen Spalten wurden sinnvolle Namen vergeben. Für eine bessere Übersicht wurde die Auswahl auf drei Parteien beschränkt: die Demokratische und die Republikanische Partei sowie die Green Party. Es gibt noch weitere Möglichkeiten zur Verarbeitung der Daten. So können zum Beispiel mathematische Formeln angewendet werden. Auch die Inhalte einzelner Zellen lassen sich anpassen.



ABC Kandidaten_Na...	ABC Partei	#,# Gesamtbeitrag	# Anzahl_Beiträge
19 Categories	4 Categories	25 - 997,534	1 - 2.35k
Carson, Benjamin S	REP	244943	1824
Cruz, Rafael Edwar	REP	348112	1764
Clinton, Hillary R	DEM	997534	898
Sanders, Bernard	DEM	217178	2352
Paul, Rand	REP	54078	274
Fiorina, Carly	REP	63046	272
Graham, Lindsey O	REP	19592	32
Christie, Christop	REP	97220	50
Walker, Scott	REP	40965	58
Bush, Jeb	REP	340381	226
Webb, James	UNK	2350	4
Kasich, John R	REP	65832	60
Perry, James R (Ri	REP	21400	13
O'malley, Martin J	DEM	43823	43
Stein, Jill	GRE	350	2
Jindal, Bobby	REP	15365	7
Santorum, Richard J.	REP	7665	10
Pataki, George E	REP	5100	3
Wells, Robert Carr	DEM	25	1

Abbildung 4.2.: Anzahl der Beiträge
Quelle: Eigene Darstellung in Cloud Dataprep

Die Verbindung dieser beiden Quellen führt zu einer Datei mit fünf Millionen Zeilen. DataPrep benutzt MapReduce (siehe Abschnitt 3.2.3) zur Verarbeitung der Daten. Abbildung 4.3 zeigt das Ergebnis, nachdem der Job und die Verarbeitung durchgeführt wurden. Dabei ist zu sehen, dass die Fehlerquote bei unter 1 % liegt. Die Ergebnisse von Dataprep können nur in Cloud Storage oder BigQuery gespeichert werden. Das Speicherformat ist beschränkt auf CSV oder JSON.

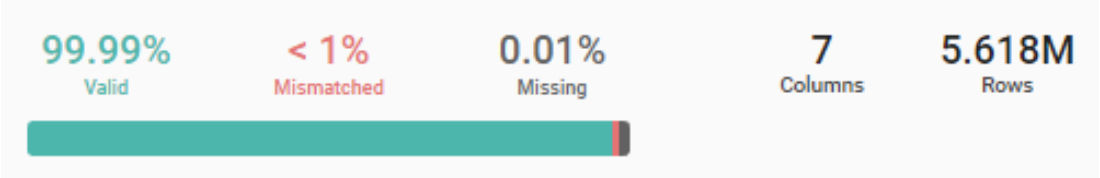


Abbildung 4.3.: Quelle: Ergebnis der Auswertung in Dataprep
Eigene Darstellung in Cloud Dataprep

Die Bearbeitung dieser Daten dauerte in Dataprep 18 Minuten. Die Daten werden von Dataprep im Batch verarbeitet. Danach sind die Daten in BigQuery verfügbar.

4.3.2. Bearbeitung mit BigQuery

BigQuery ist ein Data Warehouse-System, das Daten im Petabyte-Bereich verwaltet und eine Analyse dieser Daten ermöglicht. Es ermöglicht eine schnelle und interaktive Verarbeitung sowie Abfragen von großen Datensätzen. BigQuery ist auf die Analyse von Milliarden von Zeilen ausgelegt. Dabei wird eine SQL-ähnliche Syntax verwendet, die bereits nahezu alle Funktionen von SQL zur Verfügung stellt und stetig weiterentwickelt wird. BigQuery ist in hohem Maße skalierbar und passt sich so den jeweiligen Anforderungen an.

Nachdem das Ergebnis in BigQuery geladen wurde, können die Daten dort weiterverarbeitet werden. BigQuery unterstützt neben dem Ergebnis aus Dataprep auch Daten aus der Google Cloud, Google Drive sowie eigenständig hochgeladene Daten. BigQuery unterstützt folgende Formate: CSV, JSON, Avro, Parquet und ORC. Abbildung 4.4 zeigt einen Ausschnitt der Tabelle. Neben der ID sind jetzt alle Spendernamen sowie der Betrag zu sehen, der von diesen Personen gespendet wurde. Weiterhin sind alle Kandidaten zu sehen, die diese Spenden erhalten haben, sowie deren Parteizugehörigkeit und der Staat bzw. die Stadt, aus denen die Spender stammen. Insgesamt hält diese Tabelle 5.617.780 Zeilen.

Row	ID	Spender	Spendensumme	Kandidat	Partei	Stadt	Staat
1	C00448514	BELZ, RONALD	1800	LEATHERWOOD, THOMAS F III	REP	ARLINGTON	TN
2	C00336388	SLIFKA, ANDREW	400	CAPUANO, MICHAEL E	DEM	SOMERVILLE	MA
3	C00561001	BERNARD, HADAS	2600	BISHOP, MIKE	REP	ROCHESTER	MI
4	C00561001	RONAN, SCOTT A	2500	BISHOP, MIKE	REP	ROCHESTER	MI
5	C00575126	SCHIFTER, MAIDA	450	RASKIN, JAMIE	DEM	TAKOMA PARK	MD
30	C00459255	PERKINS, JUDITH	1944	YOUNG, TODD CHRISTOPHER	REP	BLOOMINGTON	IN
31	C00459255	SIMMERMAN, KYLE A	1250	YOUNG, TODD CHRISTOPHER	REP	BLOOMINGTON	IN
32	C00459255	HINGSON, CONSTANTINE	1500	YOUNG, TODD CHRISTOPHER	REP	BLOOMINGTON	IN
33	C00459255	HAYS, MISTI,	1400	YOUNG, TODD CHRISTOPHER	REP	BLOOMINGTON	IN
34	C00459255	BROWN, FRANCES C MRS	190	YOUNG, TODD CHRISTOPHER	REP	BLOOMINGTON	IN

Table JSON First < Prev Rows 1 - 34 of 5617780 Next > Last

Abbildung 4.4.: Übersicht über alle Spenden
Quelle: Eigene Darstellung in BigQuery

Die Daten erlauben verschiedene Abfragen. Dabei sollte das Ziel grundsätzlich der Gewinn von neuen Informationen sein. Ein Beispiel für eine Abfrage wird im Folgenden aufgezeigt. Abbildung 4.4 zeigt die Anzahl aller Spenden, die auf einen bestimmten Kandidaten entfallen sind. Gesucht wird nach einem Kandidaten, der in einem Feld den Wert ‚Clinton' enthält.

```
1  #Zeige die Anzahl aller Spenden für einen Kandidaten
2  SELECT
3    Kandidat, Count(Spendensumme) as Anzahl_der_Spenden
4  FROM
5    [snappy-nomad-168008:Kampagnebeitraege.Kampagnenbeitraege_Spender]
6  Where
7       kandidat CONTAINS 'CLINTON'
8  GROUP BY
9      Kandidat
```

Abbildung 4.5.: Zeigt den Code für die Anzahl der Spenden für einen Kandidaten
Quelle: Eigene Darstellung in BigQuery

Abbildung 4.6 zeigt, dass nur ein Kandidat gefunden wurde. In diesem Fall enthält das Feld jedoch zwei Einträge, was darauf zurückzuführen ist, dass die beiden hier aufgeführten Personen zusammengearbeitet haben. Insgesamt haben beide 2.515.636 Spenden erhalten. Das Ausführen dieser Abfrage nimmt in BigQuery 1,1 Sekunden in Anspruch.

Row	Kandidat	Anzahl_der_Spenden
1	CLINTON, HILLARY RODHAM / TIMOTHY MICHAEL KAINE	2515636

Abbildung 4.6.: Anzahl der Spenden für einen Kandidaten
Quelle: Eigene Darstellung in BigQuery

Ein weiterer interessanter Punkt ist die Höhe der Spendengelder, die die einzelnen Kandidaten erhalten haben. Im gleichen Schritt ist zu sehen, welcher Kandidat die meisten Spenden erhalten hat. Abbildung 4.7 zeigt den Code. Dabei werden zunächst die drei Spalten ausgewählt. Danach wird angegeben, dass die Kandidaten und die zugehörigen Parteien gruppiert werden sollen, sodass für jeden Kandidaten die Summe der Spenden angezeigt wird. Zum Schluss werden die Ergebnisse in absteigender Reihenfolge angezeigt. Zur besseren Übersicht werden nur die Top 10 ausgegeben.

```
SELECT
  kandidat,
  SUM(Spendensumme) AS Gesamt_Spenden,
  Partei
FROM
  [snappy-nomad-168008:Kampagnebeitraege.Kampagnenbeitraege_Spender]
GROUP BY
  kandidat,
  Partei
ORDER BY
  Gesamt_Spenden DESC LIMIT 10;
```

Abbildung 4.7.: Summiert alle Spenden für die Top 10 Kandidaten
Quelle: Eigene Darstellung in BigQuery

Aus Abbildung 4.8 kann entnommen werden, welcher Kandidat am meisten Spenden im Jahr 2016 erhalten hat. Ferner ist daran abzulesen, welche Parteien am stärksten vertreten waren. So haben Hillary Clinton und Timothy Kaine mit 292.874.290 $ am meisten Spenden erhalten.

Row	kandidat	Gesamt_Spenden	Partei
1	CLINTON, HILLARY RODHAM / TIMOTHY MICHAEL KAINE	292874290	DEM
2	SANDERS, BERNARD	84444828	DEM
3	CRUZ, RAFAEL EDWARD "TED"	64059158	REP
4	BUSH, JEB	34320240	REP
5	CARSON, BENJAMIN S SR MD	29051368	REP
6	KASICH, JOHN R	15170370	REP
7	YOUNG, TODD CHRISTOPHER	13964704	REP
8	MURPHY, PATRICK E	10190786	DEM
9	CHRISTIE, CHRISTOPHER J	8251785	REP
10	BLUNT, ROY	7614135	REP

Abbildung 4.8.: Top 10 Spendenkandidaten mit ihrer Partei
Quelle: Eigene Darstellung in BigQuery

BigQuery ermöglicht es, komplexe Abfragen in SQL zu schreiben, sodass Ergebnisse innerhalb von wenigen Sekunden verfügbar sind – auch wenn eine Tabelle über mehrere Millionen Zeilen verfügt. Unternehmen können große Datenmengen mit BigQuery auswerten. Im vorliegenden Fall könnten Unternehmen etwa herausfiltern, in welchen Staaten am wenigsten gespendet wurde, um in diesen dann mehr Werbung zu machen.

4.3.3. Visualisierung mit Data Studio

Zur Visualisierung der Daten bietet Google die Anwendung Data Studio an. Mit dieser Anwendung können Unternehmen analysierte Daten nutzen, um aus diesen neue Erkenntnisse zu gewinnen, was letztendlich zu einer Erhöhung des Unternehmenswertes führt.

Zur Visualisierung kann neben den Tabellen aus BigQuery auch auf andere Quellen zurückgegriffen werden. Als Quellen können zum Beispiel Google Storage, Cloud SQL, MySQL, PostgreSQL, YouTube Analytics und eigene Daten benutzt werden. Für die Bearbeitung können nicht nur Tabellen aus BigQuery verwendet werden, sondern auch selbst erstellte Views. Diese beinhalten spezifische Daten, die für die Auswertung exakter sein können. Diese können mit Data Marts (siehe Abschnitt 2.4.2) verglichen werden, da auch diese einen Ausschnitt der Datenbestände enthalten.

Im Beispiel wird eine Tabelle und eine View verwendet. Beide dienen zur Auswertung der Spenden für die Kandidaten im Jahr 2016. Ziel dabei ist die Visualisierung sowie der Gewinn von Informationen durch ein Zusammenführen von Daten. Beide Quellen werden in Data Studio geladen. In einem nächsten Schritt können verschiedene Visualisierungen vorgenommen werden. Dabei ist grundsätzlich darauf zu achten, dass die ausgewählten Visualisierungen auch den Sachstand widerspiegeln.

In der fertigen Visualisierung, die in Abbildung 4.9 zu sehen ist, sind alle maßgeblichen Informationen auf einem Bildschirm erfasst. Auf der linken Seite sind alle Daten für die Kandidaten und die erhaltenen Spenden aufgeführt. Dabei wird zunächst die Höhe der Spendensumme aufgeführt, die ein Kandidat erhalten hat. Die Sortierung orientiert sich an der Höhe der Spenden in absteigender Reihenfolge. Es ist sofort ersichtlich, dass Clinton und Kaine mit Abstand die größte Summe an Spenden erhalten haben.

Danach wird mit einem Kreisdiagramm veranschaulicht, welche Personen oder Organisationen insgesamt am häufigsten gespendet haben. Zu sehen ist, dass Priscilla Gilman mit einer Gesamtzahl von 2.097 die meisten Spenden getätigt hat. Auf dem nächsten Rang befindet sich die Organisation ActBlue mit 2 079 Spenden. Insgesamt haben 1.039.408 Personen oder Organisationen im Jahr 2016 für die Personen im Wahlkampf gespendet. Dabei kam ein Gesamtbetrag von 1.024.314.140 $ zusammen.

Auf der rechten Seite sind alle Informationen über die Herkunft der Spenden verzeichnet. Für eine bessere Übersicht wurde ein Teil der Karte ausgewählt, der einen Ausschnitt von Amerika zeigt. Die auf der Karte verteilten kleinen Kreise veranschaulichen die jeweilige Anzahl der Spender. Je blauer ein Kreis ist, desto größer ist die Anzahl der Spenden. Interessant ist, dass die meisten Spenden aus dem Osten stammen. Im Norden sind dagegen kaum Spenden eingegangen. Aus dieser Grafik lassen sich Informationen darüber ableiten, in welchen Gebieten mehr Wahlkampf nötig wäre. Zudem lässt sich aus dieser Grafik ableiten, in welchen Gebieten sich der Wahlkampf am meisten lohnt, weil dort die meisten Personen zu spenden bereit sind. Die untenstehende Tabelle zeigt noch einmal eine Auflistung aller Städte sowie die Höhe der Spendengelder. Die Tabelle erlaubt eine schnelle Übersicht über die Städte, aus denen die meisten Spendengelder stammen.

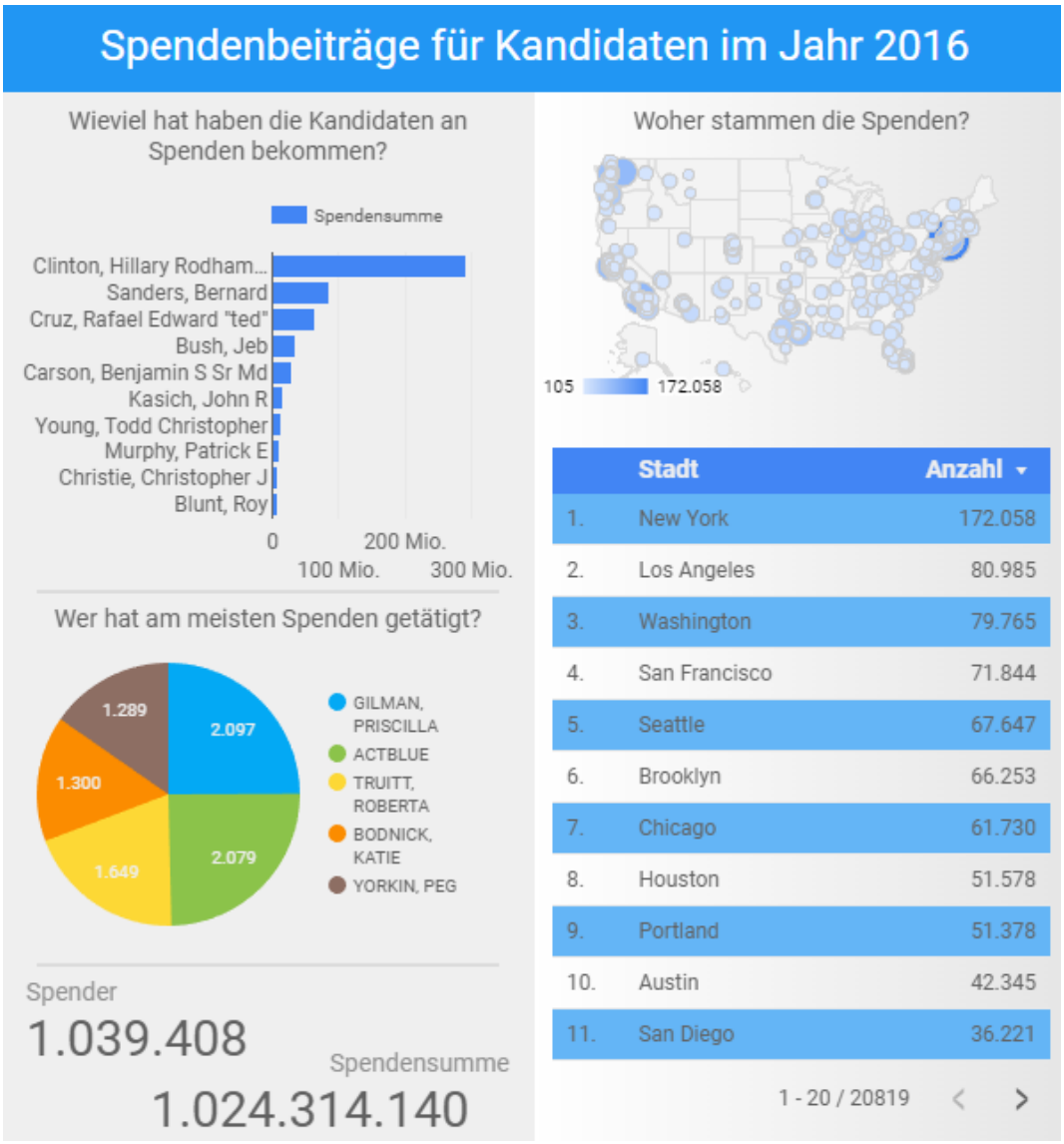


Abbildung 4.9.: Übersicht der Visualisierung
Quelle: Eigene Darstellung in DataStudio

Um eine bessere Übersicht zu erhalten, gibt es die Möglichkeit, mit den Kreisen auf der Karte zu interagieren. Auf diese Weise lässt sich ein Ort mit vielen Spenden noch einmal in eine kleinere Übersicht aufteilen. Somit lassen sich auch kleinere Städte berücksichtigen. Eine weitere Möglichkeit zur besseren Übersicht bieten die Diagramme. Zwar ist so auf den ersten Blick zu sehen, welcher Kandidat die höchste Spendensumme erhalten hat, aber es fehlt die Information

über die exakte Zahl. Weitere Informationen werden angezeigt, wenn die Balken markiert werden. Daraus ist ersichtlich, welche Spendensumme ein Kandidat exakt bekommen hat. Diese Information ist für eine schnelle Übersicht jedoch überflüssig und würde die Gestaltung der Seite stören, da diese dann mit Daten und zahlen überladen wäre.

4.4. Experimentelle Untersuchung: Microsoft Azure

Als Referenzanwendung wird Microsoft Azure untersucht. Die Daten zur Auswertung sind wieder die in Kapitel 4.2.1 vorgestellten Daten. Die Daten liegen in ihrer Rohform vor. Microsoft Azure biete viele Anwendungen zur Umsetzung an. Im vorliegenden Fall werden drei dieser Anwendungen herangezogen. Die Daten werden in Azure Blob Storage gespeichert und verwaltet, durch Azure Machine Learning verarbeitet und mit Power BI visualisiert.

4.4.1. Speicherung mit Azure Blob Storage

Beide Daten werden in Azure Blob Storage geladen. Dieser bietet Kapazitäten im Exabyte-Bereich an. Für Big Data bietet sich eine große Cloud-Plattform an. Die Daten werden als Blob gespeichert. Blob steht für Binary Large Objects und beschreibt binäre Datenobjekte, die in Datenbanken gespeichert werden können. Dabei werden die Daten in 100 MB große Blöcke aufgeteilt. Mit Block-Blobs können große Blobs effizient hochgeladen werden. Blockblobs bestehen aus Blöcken, die jeweils durch eine Block-ID identifiziert werden. Die maximale Größe eines Block-Blobs liegt bei etwas mehr als 4,75 TB (100 MB x 50.000 Blöcke) (Shahan und Myers, 2018). Die Daten können für die weitere Verarbeitung aus Azure Blob Storage geladen und gespeichert werden. In einem ersten Schritt werden die Daten für die Bereinigung in Azure Machine Learning geladen. Nach der Bereinigung werden die Ergebnisse wieder in Azure Blob Storage gespeichert. Dann werden diese Daten von Power BI geladen und für die Visualisierung aufbereitet.

4.4.2. Bereinigung der Daten mit Azure Machine Learning Studio

Azure Machine Learning Studio ist eine Entwicklungsumgebung zur Erstellung und Operationalisierung des Machine Learning Workflows auf Azure. Mit dieser Umgebung können Daten aber auch bereinigt und transformiert werden. Weiterhin lassen sich auch Vorhersageanalysen erstellen. Die Funktionen und Parameter können iterativ angepasst werden, bis ein effektives Modell erreicht ist. Für die Umsetzung wird nur der Teil zur Säuberung und Transformationen von Azure Machine Learning genutzt. Für die praktische Anwendung werden die Daten zunächst aus der Quelle extrahiert. Danach werden die Daten geladen und transformiert (siehe

Abschnitt 3.3.1). Die Daten werden in ihrer Rohform aus dem Blob Storage geladen und danach transformiert. Es gibt viele weitere Quellen, die von Azure Machine Learning genutzt werden. Die Daten können aus Azure SQL Database, Hive Query, Azure DocumentDB oder Web URL via HTTP stammen.

Im ersten Schritt werden die beiden Quelldaten geladen. Danach werden beide Daten miteinander verbunden. Nur so können Abfragen zu Kandidaten und Spenden gestellt und das Ergebnis anschließend visualisiert werden. Die Daten beinhalten viele Spalten, die nicht für die Bearbeitung notwendig sind und sich daher entfernen lassen. Zum Schluss werden alle leeren Felder entfernt. Die in Abbildung 4.10 dargestellten Schritte bilden daher nur einen Teil des vollständigen Experiments.

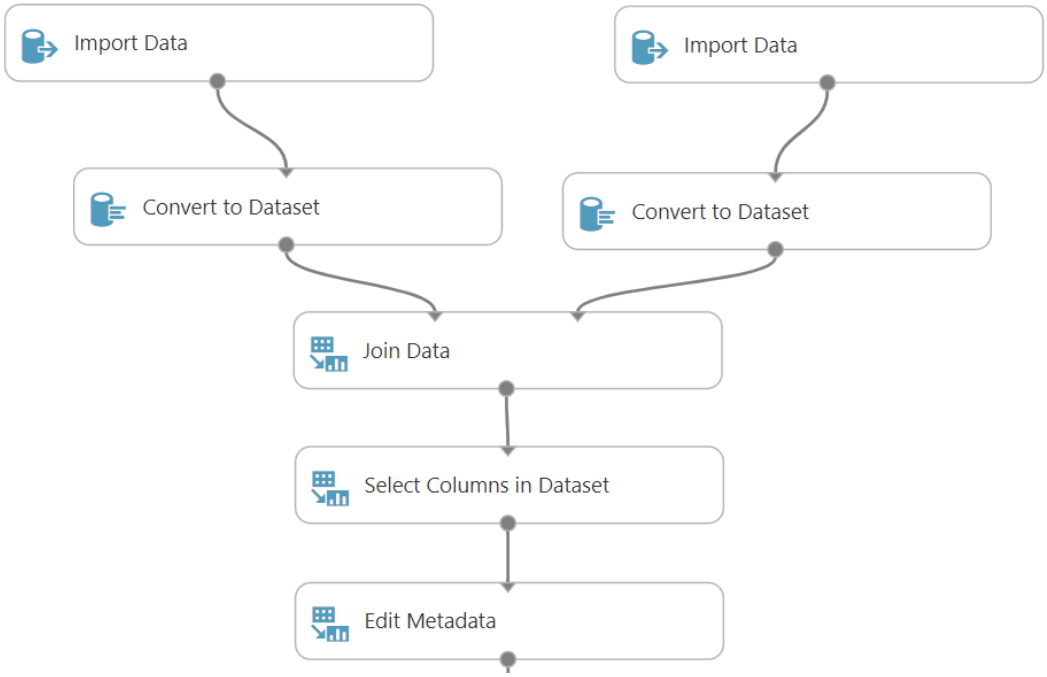


Abbildung 4.10.: Import und Join der Datenquellen
Quelle: Eigene Darstellung in Machine Learning Studio

Azure Machine Learning bietet SQL zur Bearbeitung an. Daher wird kein zusätzliches Programm – wie Dataprep bei Google (siehe Abschnitt 4.3.2) – benötigt, um die Daten zu bearbeiten. In diesem Schritt werden SQL-Abfragen erstellt, die die Daten weiter transformieren. Dabei sollen nur positive Spendensummen erfasst werden. Weiterhin werden zur besseren Übersicht nur

drei Parteien ausgewählt. Abbildung 4.11 stellt einen weiteren Ausschnitt dar. Hier werden die Daten automatisch in eine CSV-Datei transformiert. Im letzten Schritt wird das Ergebnis wieder in den Blob Storage exportiert.

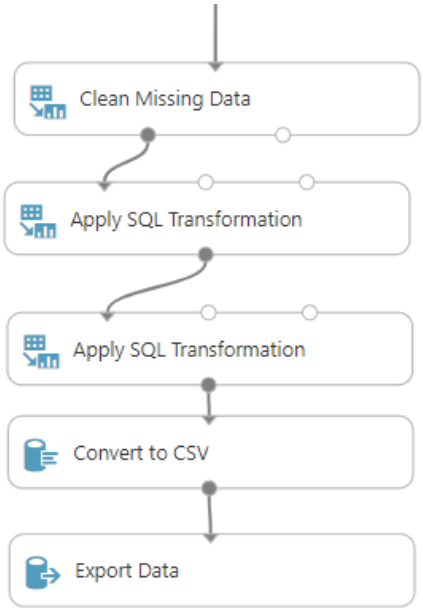


Abbildung 4.11.: SQL und Export der Datei
Quelle: Eigene Darstellung in Machine Learning Studio

Azure Maschine Learning bietet umfangreiche Funktionen zur Transformierung und Säuberung von Rohdaten. Weiterhin bietet die Anwendung Möglichkeiten zur Textanalyse sowie zum Machine Learning und Unterstützt R und Python. Für das Data Mining (siehe Abschnitt 2.5.1) stellt Azure ML Studio eine große Anzahl von Algorithmen zum maschinellen Lernen bereit. Mit diesen Komponenten lassen sich analytische Experimente durchführen.

4.4.3. Visualisierung mit Microsoft Power BI

Die bereinigten Daten aus Abschnitt 4.4.2 können jetzt aus dem Blob Storage geladen werden. Wie in Abschnitt 2.5.3, wird auch an dieser Stelle auf das von Microsoft angebotene Tool Power BI zurückgegriffen. Das Tool bietet eine direkte Verbindung und erlaubt es somit, die Daten direkt aus dem Blob Storage in Power BI zu laden.

In Abbildung 4.12 wurde beispielhaft Hillary Clinton selektiert. Auf diese Weise lassen sich alle Informationen zu diesen Kandidaten präsentieren: Welche Spendengelder diese Person

erhalten hat, aus welchen Städten die Spenden stammen und aus wie vielen Einzelspenden sich die Summe zusammensetzt. Weiterhin zeigt das Kreisdiagramm, welche Personen am meisten gespendet haben. Im vorliegenden Beispiel hat Peg Yorgin 1288 Mal gespendet. Dabei gingen nur acht Spenden an den betreffenden Kandidaten.

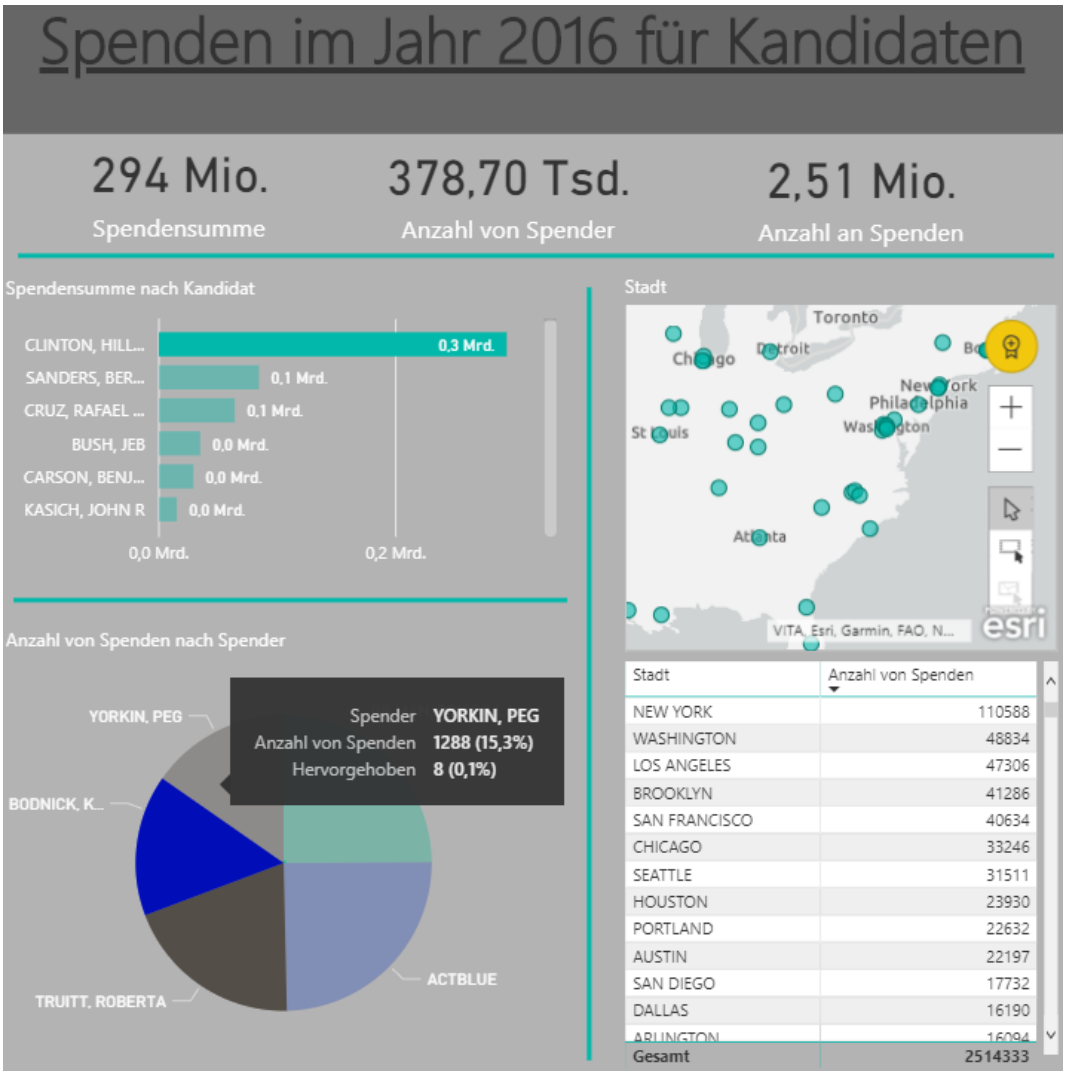


Abbildung 4.12.: Power BI Dashboard mit Auswahl von Hillary Clinton
Quelle: Eigene Darstellung in Power BI

Das Ergebnis zeigt die Eigenschaften von Dashboards (siehe Abschnitt 2.5.3). Alle Informationen sind auf einer Seite zu sehen. Die Aufteilung zeigt dem Benutzer schnell, wo die Informatio-

nen zu finden sind. Für Unternehmen bieten Visualisierungen die Möglichkeit, mit großer Geschwindigkeit Informationen aus großen Datenmengen zu gewinnen.

4.5. Experimentelle Untersuchung: Amazon Web Services (AWS)

Als drittes Referenzsystem wird AWS verwendet. Die auszuwertenden Informationen bilden dabei wiederum die in Kapitel 4.2.1 vorgestellten Daten. Es werden nur einige Anwendungen von Amazon für die Umsetzung verwendet. Amazon bietet kein kostenloses Kontingent für alle Anwendungen an, sondern für jede Anwendung individuelle Angebote. Für diese Umsetzung wurde Amazon S3 zur Speicherung genutzt, das ein Kontingent von 5 GB bereitstellt. AWS Glue, das eine Speicherung von einer Million Objekten im Datenkatalog erlaubt, wurde zur Bearbeitung, Säuberung und Transformierung der Daten herangezogen. Zur Visualisierung wird Amazon QuickSight genutzt, das 1 GB Kapazität zur Verfügung stellt. Viele Anwendungen sind nicht kostenlos enthalten. So können bei AWS nicht alle Produkte getestet werden.

4.5.1. Speicherung mit AWS S3

Beide Daten werden in AWS S3 geladen. S3 bietet im Rahmen der kostenlosen Nutzung hinreichende Kapazitäten für die Umsetzung an. Ziel von Amazon S3 ist eine 99,9-prozentige Verfügbarkeit der Daten. Der Preis passt sich an den tatsächlich genutzten Speicherplatz an und ist somit skalierbar. Auf diese Weise lässt sich der Dienst ebenso von kleinen wie auch von großen Unternehmen gleichermaßen nutzen. Die Verbindung zu anderen Anwendungen innerhalb von AWS funktioniert ohne Probleme. Wieder werden die Daten als Datenquellen genutzt und bereitgestellt, damit diese dann in weiteren Schritten verarbeitet werden können. Auch in diesem Fall werden die Ergebnisse der Transformationen in S3 gespeichert.

4.5.2. Bereinigung der Daten mit AWS Glue

Mit AWS Glue lässt sich eine Datenbereinigung durchführen. Als Quelle können die Daten aus Amazon S3 oder DynamoDB stammen. Zuerst muss eine Datenbank erstellt werden. In dieser können dann Tabellen gespeichert werden. Für das Erstellen von Tabellen kann in Glue ein Crawler benutzt werden. Dieser identifiziert die Metadaten einer Datei und erstellt Tabellendefinitionen. Sobald alle Daten erfasst worden sind, kann ein ETL-Job erstellt werden (siehe Abschnitt 2.3.2). Hierbei werden die Quelldaten und der Zielort ausgewählt sowie die

Transformation. In Abbildung 4.13 werden die Rohdaten auf der linken und die zu erhaltenden Daten auf der rechten Seite aufgeführt.

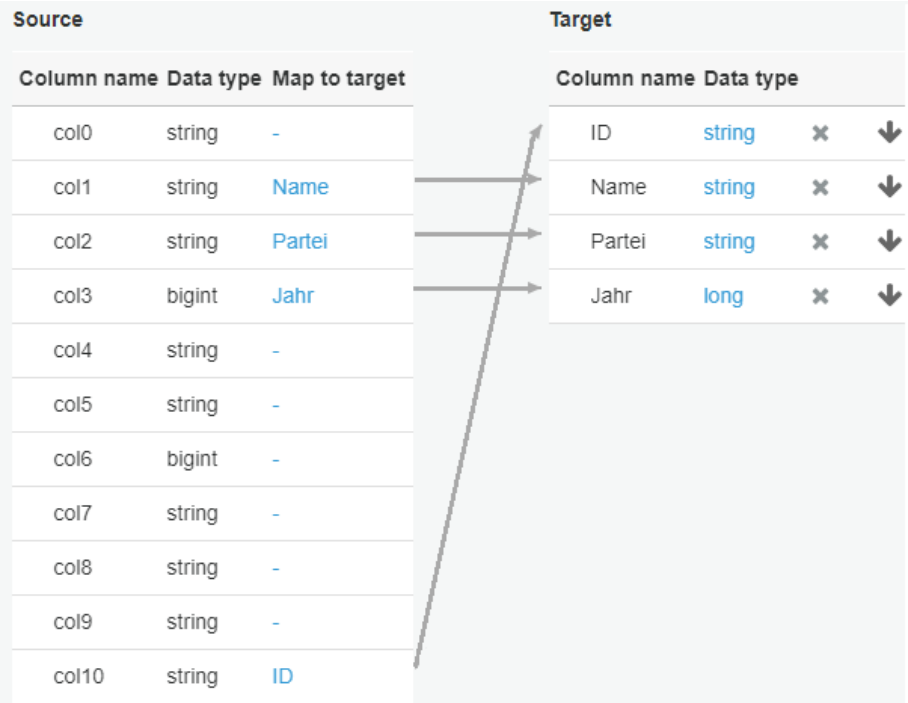


Abbildung 4.13.: Transformation der Quelldaten in das Zielmuster
Quelle: Eigene Darstellung in AWS Glue

Im letzten Schritt können noch Änderungen vorgenommen werden: Die Daten können noch angepasst und das Skript verändert werden. Nach dem Ausführen wird das Ergebnis als CSV-Datei in Amazon S3 gespeichert. Weitere Speicherformate, die Glue anbietet, sind: JSON, Parquet, ORC und Avro. Die Daten können jetzt in einer Datenbank verarbeitet werden.

4.5.3. Visualisierung mit Amazon QuickSight

Für den Import aus S3 benötigt QuickSight die Berechtigung für S3 Bucket. Für das Laden wird eine JSON-Datei benötigt. Diese enthält neben dem Dateipfad weitere optionale Einstellungen. Es können dabei das Format angeben werden, das Feldtrennzeichen der Datei sowie der Umstand, ob die Datei eine Kopfzeile enthält. Weitere Quellen sind u. a. CSV, JSON, xlsx, Amazon S3, AWS RDS, MySQL, Twitter und Redshift. Leider wird für die Bearbeitung der Ergebnisse aus AWS Glue mehr als der zur Verfügung gestellte Speicherplatz von 1 GB benötigt. Daher wurde an dieser Stelle das Ergebnis der Bearbeitung mit Google Dataprep (siehe Abschnitt

4.3.1) weiterverwendet. Diese beinhaltet einen Verbund aus den Kandidaten und Spendern. Das Ergebnis wird über das Manifest aus S3 in QuickSight geladen. Die Visualisierung in Abbildung 4.14 basiert auf ähnlichen Informationen wie diejenigen in Abschnitt 4.3.3 und 4.4.3.

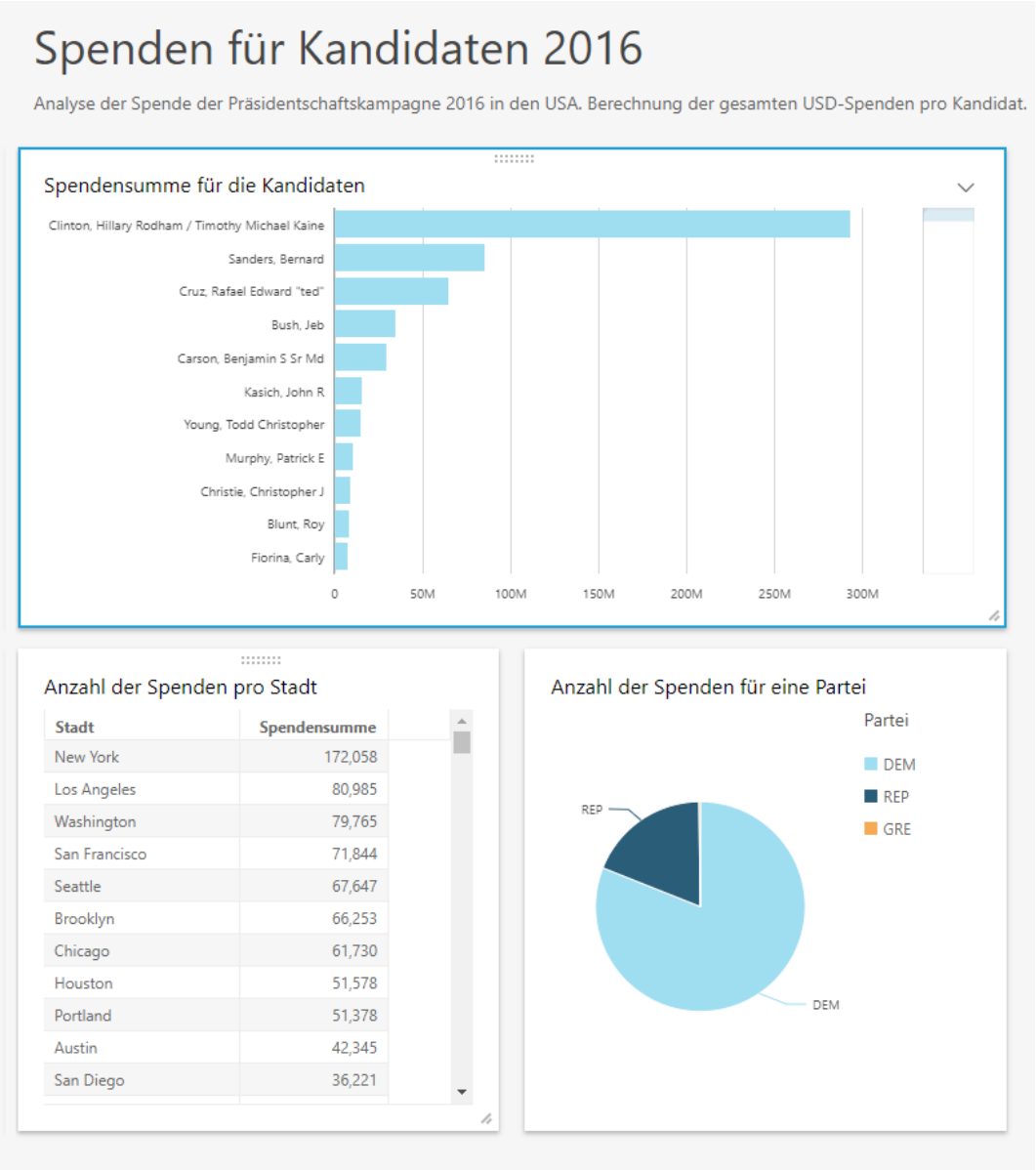


Abbildung 4.14.: Dashboard in AWS QuickSight
Quelle: Eigene Darstellung in AWS QuickSight

Bei der Umsetzung konnten nicht alle Anforderungen erfüllt werden. Anpassungen am Design waren nicht weiter möglich. Es konnte keine Farbanpassung beim Hintergrund oder zur Aufteilung des Dashboards vorgenommen werden. Eine Darstellung der Weltkarte war mit QuickSight nicht möglich, da die Stadtnamen nicht als String erkannt wurden. Für die Umsetzung werden geografische Daten benötigt. Die Darstellung einer Karte mit den Stadtnamen bringt bei Power BI und Google Data Studio keine Probleme mit sich (siehe Abschnitt 4.3.3 und 4.4.3).

Mit QuickSight können zum Beispiel auch OLAP-Operationen durchgeführt werden. Wenn eine Produktfamilie in der Visualisierung vorhanden ist, kann zum Beispiel ein Drill-down (siehe Abschnitt 2.5.2) durchgeführt werden. Auf diese Weise lassen sich Informationen über eine darunterliegende Dimensionshierarchie erlangen. So können weitere Informationen zu dieser Produktfamilie dargestellt werden – zum Beispiel einzelne Typen.

4.6. Evaluation

In diesem Kapitel werden die Konzepte aus Kapitel 2 und 3 sowie die Realisierung aus Kapitel 4 abgeglichen und einer kritischen Betrachtung unterzogen.

Viele Cloud-Anbieter haben Lösungen für die neuen Herausforderungen von Big Data entwickelt und bieten diese für Unternehmen an. Die Anwendungen sind gut dokumentiert und ermöglichen einen schnellen Einstieg in die jeweilige Infrastruktur. Für die Umsetzung wurden drei Anwendungen von drei unterschiedlichen Anbietern miteinander verglichen. Alle Anbieter bieten neben den ausgewählten Anwendungen eine weitere große Anzahl von Anwendungen für unterschiedliche Lösungen an.

Für die Verarbeitung, Säuberung und Transformation bieten alle drei Anbieter Lösungen an. Microsoft Azure bietet den größten Funktionsumfang. Neben einer Säuberung und Transformation bieten sich hier auch Möglichkeiten zum Machine Learning (siehe Abschnitt 4.4.2). Google Cloud Dataprep kann zur Vorverarbeitung von Rohdaten verwendet werden. Das Programm stellt viele Filter und Modifikationsmöglichkeiten bereit. Die geringsten Möglichkeiten bietet Amazon mit Glue. Diese Anwendung ist lediglich dazu in der Lage, Metadaten aus den Rohdaten zu gewinnen, die dann zur Transformation genutzt werden können. Für Glue wird mehr Vorwissen benötigt, da es keine einfache Bedienoberfläche zur Bearbeitung gibt, sondern

alle Filter selbst geschrieben werden müssen, wodurch sich der Einstieg beschwerlich gestaltet.

DataStudio und Power BI bieten die umfangreichsten Visualisierungsmöglichkeiten. Das Dashboard lässt sich schnell einrichten und vielseitig anpassen. Dabei können die Daten aus unterschiedlichen Quellen stammen. Auch die Verwendung von unterschiedlichen Quellen in einer Übersicht ist möglich, und bietet eine größere Vielfalt der Informationsgewinnung. Amazon QuickSight bietet weniger Anpassungsmöglichkeiten. Auch gestaltet sich die Verarbeitung von Daten nicht so einfach wie bei Google und Microsoft Azure.

Die interne Verarbeitung der Daten sowie das Importieren und Exportieren der bearbeiteten Daten ging bei allen Anbietern effizient vonstatten. Der Speicherplatz in der Cloud war für den Test bei allen Anbietern ausreichend. Die Daten konnten sowohl in Rohform als auch bearbeitet am gleichen Platz in der Cloud abgespeichert werden.

Eine Bewertung der Anwendungen wird nach der Umsetzung in Tabelle 4.2 dargestellt. Das Bewertungsschema orientiert sich an den Schulnoten. Die Preise variieren jedoch je nach Region und Größe des genutzen Speicherplatzes.

In der Tabelle werden die drei Anbieter miteinander verglichen. Dabei werden sowohl die allgemeinen Kosten als auch die Kosten für den Speicherplatz aufgelistet. Zudem finden sich Angaben zu Benutzbarkeit, Umfang und Besonderheiten der einzelnen ETL-Anwendungen. Am Ende werden die Visualisierungen mit dem Dashboard verglichen und bewertet. Für die Umsetzung entstehen die folgenden Kosten:

- **Google Cloud:** 1,63€
- **Microsoft Azure:** 1,82€
- **Amazon Web Services:** 5,12€

	Google Cloud	Microsoft Azure	Amazon Web Services
Allgemein			
Umfang	unbegrenzt bis Guthaben von 256,70€ aufgebraucht	unbegrenzt bis Guthaben von 170€ aufgebraucht	kein Guthaben; Nutzung begrenzt auf ausgewählte Anwendungen
Speicher			
Einstieg	5 GB/Monat kostenlos	5 GB/Monat kostenlos	5 GB/Monat kostenlos
Ab 5GB	0.022€ pro GB/Monat	0,019€ pro GB/Monat	0,020€ pro GB/Monat
ETL			
Kosten	0,08€ pro Stunde	0,84€ pro Stunde	0,38€ pro Stunde
Benutzbarkeit	1	1	3
Umfang	2	1	3
Besonderheiten	Untersuchung, Bereinigung und Vorbereitung von Daten	Transformation von Daten. SQL-Abfragen. R und Python unterstützt. Machine Learning.	ETL-Service
Dashboard			
Kosten	kostenlos	kostenlos bis 8,40€ pro Benutzer/Monat	1 GB Kapazität kostenlos, danach 0,22€ pro GB/Monat
Benutzbarkeit	2	2	4
Umfang	2	1	4

Tabelle 4.2.: Vergleich der Kosten der Anbieter

Nachteile ergeben sich vor allem im Bereich der Sicherheit. Die Cloud-Anbieter müssen einen großen technischen Aufwand betreiben, um Datensicherheit gewährleisten zu können. Hacker versuchen personenbezogene Nutzerdaten von Servern zu stehlen. Der Datenschutz ist für Unternehmen von großer Bedeutung: Jedes Unternehmen muss daher prüfen, wie relevant die jeweiligen Daten sind. Könnten die Daten personenbezogene Daten beinhalten, dürfen diese Daten weitergegeben werden und was passiert, wenn diese Daten von Unbefugten gelesen oder kopiert werden? Die Wahrscheinlichkeit bei einer eigenen Infrastruktur ist in dieser Hinsicht deutlich geringer als bei Cloud-Anbietern. Ein Unternehmen hat die Möglichkeit, auf die eigene Infrastruktur Einfluss zu nehmen, was bei Cloud-Anbietern nicht möglich ist. Zwar gelten auch hier die EU-Datenschutzrichtlinien. Doch diese relativieren sich insbesondere dann, wenn Cloud-Anbieter ihren Firmensitz und ihre Server in nicht-europäischen Ländern

haben.

Ein weiterer Nachteil für Unternehmen stellt die Abhängigkeit vom Anbieter dar. Dieser kann sich zum Beispiel unzureichend um Kunden kümmern, nicht genug Kapazitäten anbieten oder durch Ausfall den Dienst vorübergehend abschalten. Unternehmen müssen daher immer entscheiden, welche Bedeutung der Erreichbarkeit der Daten zukommt und wie hoch die Anforderungen an die Kapazität sind. Auch die IT-Kompetenz spielt eine Rolle bei der Entscheidung. Unternehmen, die über kein entsprechendes Fachpersonal verfügen, sind auf die Leistung des Cloud-Anbieters angewiesen. Auch kann es zu Problemen kommen, wenn die eigene Software an die der Cloud-Anbieter angepasst werden muss. Nicht immer passen beide zusammen, sodass die verwendete Software entsprechend ausgewechselt werden muss.

Mithilfe der Cloud erweist es sich als unproblematisch, die ersten Schritte im Big Data-Umfeld zu gestalten. Die Anwendungen sind für Einsteiger gut dokumentiert, bieten aber umfassende Komplexität, sodass auch große Unternehmen damit Informationen aus Daten gewinnen können. Die Kosten beschränken sich auf den Umfang der tatsächlichen Nutzung. Für Unternehmen sind Cloud Services im Bereich von CRM, CMC, Data Warehouse und Datenspeicherung beliebt. Die Nachteile hängen vom Einsatzbereich ab und müssen für jeden Fall neu ausgewertet werden.

5. Zusammenfassung und Ausblick

In diesem Kapitel werden der Inhalt dieser Arbeit zusammengefasst und die zentralen Ergebnisse dargestellt. Weiterhin wird ein Ausblick gegeben, wie sich die Data Warehouse-Systeme in Zukunft entwickeln werden.

5.1. Zusammenfassung

Die Speicherung und Auswertung von Daten zur Informationsgewinnung ist für Unternehmen immer schon relevant gewesen. Data Warehouse-Systeme haben Unternehmen in diesem Zusammenhang unterstützt. In Kapitel 2 wurden die klassischen Data Warehouse-Systeme erläutert. Dabei wurden einzelne Aspekte der Umsetzung erklärt.

Die Anforderungen haben sich in den letzten Jahren verändert. So sind nicht nur die Datenmenge und die Datengrößen angestiegen. Auch die Frequenz, mit der die Daten erzeugt werden, hat sich erhöht – sei es bei der Auswertung von Daten über das Wetter, Sensoren in Autos oder Social Media. Diese beträchtliche Steigerung der Last führte dazu, dass hinsichtlich der Data Warehouse-Systeme Anpassungen vorgenommen werden mussten. Viele neue Technologien wurden entwickelt. Zum Teil wurden allerdings auch völlig neue Ansätze geschaffen, um diese Herausforderungen meistern zu können. In Kapitel 3 wurde der Data Lake vorgestellt. Dieser kann als Erweiterung für ein Data Warehouse angesehen werden. Diese Systeme eröffnen eine andere Herangehensweise an den Umgang mit großen Daten.

Data Lake-Systeme lassen sich auf unterschiedliche Weise umsetzten, wobei die Kosten bei der Umsetzung eine große Rolle spielen. Zu den Kosten für die benötigte Infrastruktur kommen dabei auch Personalkosten. Bei einer gänzlich eigenständigen Umsetzung wird eine Infrastruktur von vielen Computern sowie entsprechendes Personal benötigt, das im Umgang mit diesen Anwendungen ausgebildet und auf diese spezialisiert ist. Bei Online-Lösungen – wie zum Beispiel den in Kapitel 4.3 vorgestellten – wird eine Infrastruktur bereitgestellt. So wird eine Skalierung möglich, durch die die Lösung an die Größe des jeweiligen Unternehmens angepasst werden kann. Es wird nur die benötigte Leistung bezahlt, doch diese kann sich mit

veränderten Herausforderungen für ein Unternehmen schnell ändern. Dieses System ermöglicht eine schnelle Bereinigung heterogener Daten sowie die Bearbeitung und Visualisierung von Informationen für Anwender in einem Unternehmen.

5.2. Ausblick

Data Warehouse-Systeme werden auch in Zukunft für Unternehmen von Bedeutung sein. Es wird weiterhin ein zentraler Aspekt bleiben, einen unternehmerischen Mehrwert aus den großen Datenmengen zu ziehen. Dieses Wissen kann von Anwendern genutzt werden, um auf diese Weise bessere Entscheidungen zu treffen. Big Data wird immer neue Herausforderungen mit sich bringen. Es wird beständig mehr Speicherplatz benötigt und die Vielfalt der Daten wächst kontinuierlich. Data Warehouse-Systeme müssen daher beständig erweitert und sinnvoll mit neuen Technologien ergänzt werden.

Die große Herausforderung stellt dabei weiterhin eine Verbindung von klassischen Data Warehouse-Ansätzen mit Big Data-Technologien dar. Um die Kosten für Unternehmen weiter zu senken, bieten sich Cloud-Lösungen an. Diese bieten sich aufgrund der ansteigenden Anforderungen an Ressourcen und dem gleichzeitigen Anstieg der benötigten Kapazitäten als ergänzende Maßnahme an.

A. Inhalt der CD

Dieser Arbeit liegt eine CD mit folgender Verzeichnisstruktur bei:

- **[Ausarbeitung]** beinhaltet diese Arbeit im PDF-Format.
- **[Literatur]** beinhaltet in dieser Arbeit verwendete Literatur.
- **[Quelldatei]** beinhaltet einen Ausschnitt der Quellen als CSV.
- **[Ergebnisse Google]** beinhaltet die Ergebnisse aus Google Cloud.
- **[Ergebnisse Azure]** beinhaltet die Ergebnisse aus Microsoft Azure.
- **[Ergebnisse AWS]** beinhaltet die Ergebnisse aus Amazon Web Services.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[Avinoam 2018] Avinoam, Roi: *ETL vs ELT: The Difference is in the How.* 2018. – Online verfügbar unter: https://blog.panoply.io/etl-vs-elt-the-difference-is-in-the-how [Abruf: 2018-06-09]

[Bauer und Günzel 2013] Bauer, Andreas ; Günzel, Holger: *Data-Warehouse-Systeme: Architektur, Entwicklung, Anwendung.* 4. Aufl. Paderborn : dpunkt.verlag, 2013. – ISBN 978-3898647854

[Bendel 2018] Bendel, Oliver: *Big Data.* 2018. – Online verfügbar unter: https://wirtschaftslexikon.gabler.de/definition/big-data-54101 [Abruf: 2018-05-19]

[Benghiat 2017] Benghiat, Gil: *The Data Lake Is A Design Pattern.* 2017. – Online verfügbar unter: https://medium.com/data-ops/the-data-lake-is-a-design-pattern-888323323c66 [Abruf: 2018-06-27]

[Brewer 2000] Brewer, Eric A.: *Towards robust distributed system.* 2000. – Online verfügbar unter: https://people.eecs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf [Abruf: 2018-05-28]

[Chamoni und Gluchowski 2016] Chamoni, Peter ; Gluchowski, Peter: *Analytische Informationssysteme.* 5. Aufl. Berlin : Springer, 2016. – ISBN 978-3-662-47762-5

[Dean und Ghemawat 2004] Dean, Jeffrey ; Ghemawat, Sanjay: MapReduce: Simplified Data Processing on Large Clusters. In: *OSDI 2004* (2004), S. 1–13. – Online verfügbar unter: http://users.cis.fiu.edu/~mrobi002/teaching/GoogleMapreduce-osdi04.pdf [Abruf: 2018-05-29]

[Dull 2015] Dull, Tamara: *Data Lake vs Data Warehouse: Key Differences.* 2015. – Online verfügbar unter: https://www.kdnuggets.com/2015/09/data-lake-vs-data-warehouse-key-differences.html [Abruf: 2018-04-15]

[Eicker 2011] Eicker, Stefan: *Repository-System*. 2011. – Online verfügbar unter: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Datenmanagement/Datenmanagement--Konzepte-des/Repository-System [Abruf: 2018-03-04]

[Fasel und Meier 2018] Fasel, Daniel ; Meier, Andreas: *Big Data - Grundlagen, Systeme und Nutzungspotenziale*. 1. Aufl. Berlin : Springer, 2018. – ISBN 978-3-658-20082-4

[Few 2013] Few, Stephen: *Information Dashboard Design: Displaying Data for At-A-Glance Monitoring*. 2. Aufl. Burlingame : Analytics Press, 2013. – ISBN 1938377001

[Firican 2017] Firican, George: *The 10 Vs of Big Data*. 2017. – Online verfügbar unter: https://tdwi.org/articles/2017/02/08/10-vs-of-big-data.aspx [Abruf: 2018-05-20]

[Freytag 2014] Freytag, Johann-Christoph: Grundlagen und Visisionen im Bereich Big Data. In: *Informatik Spektrum* 37 (2014), Nr. 2, S. 97–104. – Online verfügbar unter: https://link.springer.com/article/10.1007/s00287-014-0771-y [Abruf: 2018-05-23]

[Gabriel u. a. 2008] Gabriel, Roland ; Gluchowski, Peter ; Dittmar, Carsten: *Management Support Systeme und Business Intelligence*. 2. Aufl. Berlin : Springer, 2008. – ISBN 978-3-540-23543-9

[Gabriel u. a. 2009] Gabriel, Roland ; Gluchowski, Peter ; Pastwa, Alexander: *Data Warehouse and Data Mining*. 1. Aufl. Dortmund : W3L, 2009. – ISBN 3937137661

[Gadatsch und Landrock 2017] Gadatsch, Andreas ; Landrock, Holm: *Big Data für Entscheider - Entwicklung und Umsetzung datengetriebener Geschäftsmodelle*. 1. Aufl. Berlin : Springer, 2017. – ISBN 978-3-658-17339-5

[Gartner 2011] Gartner: *Big Data*. 2011. – Online verfügbar unter: https://www.gartner.com/it-glossary/big-data/ [Abruf: 2018-05-20]

[Gilbert und Lynch 2002] Gilbert, Seth ; Lynch, Nancy: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. In: *ACM SIGACT* 33 (2002), Nr. 2, S. 51–59. – Online verfügbar unter: https://dl.acm.org/citation.cfm?id=564601 [Abruf: 2018-05-28]

[Gimpel u. a. 2018] Gimpel, Henner ; Schmied, Fabian ; Stöber, Anna-Luisa: Der unbekannte Kunde – Potenziale der Integration von Kundendaten. In: *HMD Praxis der Wirtschaftsinformatik* 55 (2018), Nr. 1, S. 91–103. – Online verfügbar unter: https://link.springer.com/article/10.1365/s40702-017-0362-x [Abruf: 2018-05-30]

[Gluchowski 2012] Gluchowski, Peter: *Data Warehouse*. 2012. – Online verfügbar unter: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Business-Intelligence/Data-Warehouse [Abruf: 2018-04-20]

[Hummeltenberg 2012] Hummeltenberg, Wilhelm: *ETL*. 2012. – Online verfügbar unter: http://www.enzyklopaedie-der-wirtschaftsinformatik.de/lexikon/daten-wissen/Business-Intelligence/ETL [Abruf: 2018-05-08]

[Hunter 2018] Hunter, Caroline C.: *Campaign finance data*. 2018. – Online verfügbar unter: https://www.fec.gov/data/ [Abruf: 2018-07-26]

[Inmon 2016] Inmon, Bill: *Data Lake Architecture: Designing the Data Lake and Avoiding the Garbage Dump*. 1. Aufl. Basking Ridge : Technics Publications, 2016. – ISBN B01DPEGSO4

[Inmon 2005] Inmon, William H.: *Building the Data Warehouse*. 4. Aufl. Indianapolis : John Wiley, 2005. – ISBN 978-0764599446

[Isele und Arndt 2016] Isele, Robert ; Arndt, Natanael: Mit semantischer Datenverwaltung Big Data in den Griff bekommen. In: *Wirtschaftsinformatik Management* 8 (2016), Nr. 4, S. 56–63. – Online verfügbar unter: https://link.springer.com/article/10.1007/s35764-016-0065-z [Abruf: 2018-05-30]

[Kemper u. a. 2010] Kemper, Hans-Georg ; Baars, Henning ; Mehanna, Walid: *Business Intelligence - Grundlagen und praktische Anwendungen*. 3. Aufl. Wiesbaden : Vieweg Verlag, 2010. – ISBN 9783834807199

[Klein u. a. 2013] Klein, Dominik ; Tran-Gia, Phuoc ; Hartmann, Matthias: Big Data. In: *Informatik Spektrum* 36 (2013), Nr. 3, S. 319–323. – Online verfügbar unter: https://link.springer.com/article/10.1007/s00287-013-0702-3 [Abruf: 2018-04-22]

[Knight 2017] Knight, Michelle: *Data Warehouse vs. Data Lake Technology: Different Approaches to Managing Data*. 2017. – Online verfügbar unter:

http://www.dataversity.net/data-warehouse-vs-data-lake-technology-different-approaches-managing-data/ [Abruf: 2018-06-19]

[König u. a. 2016] König, Christian ; Schröder, Jette ; Wiegand, Erich: *Big Data - Chancen, Risiken, Entwicklungstendenzen*. 1. Aufl. Berlin : Springer, 2016. – ISBN 978-3-658-11588-3

[Lackes 2018] Lackes, Richard: *Data Mining*. 2018. – Online verfügbar unter: https://wirtschaftslexikon.gabler.de/definition/data-mining-28709 [Abruf: 2018-05-11]

[Mathis 2017] Mathis, Christian: Data Lakes. In: *Datenbank-Spektrum* 17 (2017), Nr. 3, S. 289–293. – Online verfügbar unter: https://link.springer.com/article/10.1007%2Fs13222-017-0272-7 [Abruf: 2018-05-30]

[Meier 2018] Meier, Andreas: *Werkzeuge der digitalen Wirtschaft: Big Data, NoSQL und Co.* 1. Aufl. Berlin : Springer, 2018. – ISBN 978-3-658-20336-8

[Meier und Kaufmann 2018] Meier, Andreas ; Kaufmann, Michael: *SQL- NoSQL-Datenbanken*. 8. Aufl. Berlin : Springer, 2018. – ISBN 978-3-662-47663-5

[Michalarias und Sümmchen 2013] Michalarias, Ilias ; Sümmchen, Markus: *Hadoop erschließt Big Data für Data Warehouses*. 2013. – Online verfügbar unter: http://www.isreport.de/news/hadoop-erschliesst-big-data-fuer-data-warehouses [Abruf: 2018-05-28]

[Müller und Lenz 2013] Müller, Roland M. ; Lenz, Hans-Joachim: *Business Intelligence*. 1. Aufl. Berlin : Springer, 2013. – ISBN 978-3-642-35559-2

[Müller 2014] Müller, Stefan: Die neue Realität: Erweiterung des Data Warehouse um Hadoop, NoSQL Co. In: *HMD Praxis der Wirtschaftsinformatik* 51 (2014), Nr. 4, S. 447–457. – Online verfügbar unter: https://link.springer.com/article/10.1365/s40702-014-0053-9 [Abruf: 2018-05-28]

[Pang und Szafron 2014] Pang, Candy ; Szafron, Duane: Single Source of Truth (SSOT) for Service Oriented Architecture (SOA). In: Franch, Xavier (Hrsg.) ; Ghose, Aditya K. (Hrsg.) ; Lewis, Grace A. (Hrsg.) ; Bhiri, Sami (Hrsg.): *Service-Oriented Computing*. Berlin : Springer, 2014, S. 575–589. – Online verfügbar unter: https://link.springer.com/chapter/10.1007/978-3-662-45391-9_50 [Abruf: 2018-08-10]

[Pettey 2015] Pettey, Christy: *Look Before Diving Headfirst Into a Data Lake.* 2015. – Online verfügbar unter: https://www.gartner.com/smarterwithgartner/look-before-diving-headfirst-into-a-data-lake-2/ [Abruf: 2018-06-07]

[Rahm u. a. 2015] Rahm, Erhard ; Saake, Gunter ; Sattler, Kai-Uwe: *Verteiltes und Paralleles Datenmanagement.* 8. Aufl. Berlin : Springer, 2015. – ISBN 978-3-642-45241-3

[Schildgen u. a. 2013] Schildgen, Johannes ; Jörg, Thomas ; Dessloch, Stefan: Inkrementelle Neuberechnungen in MapReduce. In: *Datenbank-Spektrum* 13 (2013), Nr. 1, S. 33–43. – Online verfügbar unter: https://link.springer.com/article/10.1007/s13222-012-0109-3 [Abruf: 2018-05-29]

[Schnider u. a. 2016] Schnider, Dani ; Jordan, Claus ; Welker, Peter ; Wehner, Joachim: *Data Warehouse Blueprints.* 1. Aufl. München : Carl Hanser, 2016. – ISBN 9783446450752

[Shahan und Myers 2018] Shahan, Robin ; Myers, Tamra: *Understanding Block Blobs, Append Blobs, and Page Blobs.* 2018. – Online verfügbar unter: https://docs.microsoft.com/en-us/rest/api/storageservices/understanding-block-blobs--append-blobs--and-page-blobs [Abruf: 2018-08-07]

[Soutier 2015] Soutier, Marius: *Von ETL zu ELT in BigData-Systemen.* 2015. – Online verfügbar unter: http://www.soutier.de/blog/2015/03/01/von-etl-zu-elt-big-data/ [Abruf: 2018-06-08]

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 27.08.2018 Jens Peter Urban