



Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Entwicklung eines Business-Intelligence-Szenarios für Fashion E-Commerce unter Verwendung von AWS-Diensten

Bachelorarbeit

Name des Studiengangs

Wirtschaftsinformatik

Fachbereich 4

vorgelegt von

Bassel Ali Naji Ghurab

Datum:

Berlin, 20.02.2025

Erstgutachter: Prof. Claßen, Ingo

Zweitgutachter: Prof. Kempa, Martin

Inhaltsverzeichnis

Abkürzungsverzeichnis:.....	5
1. Einleitung.....	6
1.1 Hintergrund und Motivation	6
1.2 Problemstellung	6
1.3 Zielsetzung der Arbeit	7
1.4 Aufbau der Thesis	8
2. Theoretischer Hintergrund	9
2.1 Grundlagen von Business Intelligence (BI)	9
2.1.1 Definition:	9
2.1.2 Hauptziele von Business Intelligence	11
2.1.3 Methoden	12
2.1.3.1 Transformationsprozess – ETL	12
2.2 Bedeutung von BI für den E-Commerce-Sektor, speziell im Fashion-Bereich	17
2.3 Überblick über Cloud-Technologien und AWS-Dienste im BI-Kontext.....	19
3 Szenario-Beschreibung und Geschäftsprozesse	22
3.1 Beschreibung der fiktiven Fashion-E-Commerce-Plattform	22
3.2 Detaillierte Darstellung der Geschäftsprozesse	22
3.2.1 Bestellprozess.....	22
3.3 Identifikation und Definition der relevanten KPI	23
3.4 Generierung der Rohdaten für die Plattform	26
3.4.1 Methodik der Datengenerierung.....	26
3.4.2 Generierung der Kundendaten (Users Table)	26
3.4.3 Generierung der Produktdaten (Products Table).....	27
3.4.4 Generierung der Bestelldaten (Orders Table)	28
3.4.5 Kritische Reflexion der Datenqualität und Limitationen.....	29
4. Einsatz von AWS-Diensten im Business Intelligence Szenario	30
4.1 Einleitung, warum AWS für Business Intelligence	30
4.2 Amazon S3 – Speicherung und Verwaltung von Daten	30
4.2.1 Grundlagen von Amazon S3	30

4.2.2 Anwendungsfall in dieser Arbeit	30
4.2.3 Vorteile von Amazon S3 in der BI-Architektur.....	31
4.3 AWS Glue – Automatisierung von ETL-Prozessen	31
4.3.1 Grundlagen von AWS Glue	31
4.3.2 Anwendung in dieser Arbeit.....	32
4.3.3 Vorteile von AWS Glue für Business Intelligence.....	32
4.3.4 Kostenübersicht	32
4.4 Amazon Athena – SQL-basierte Analyse von S3-Daten	32
4.4.1 Grundlagen von Amazon Athena	32
4.4.2 Anwendung in dieser Arbeit.....	33
4.4.3 Vorteile und Limitationen von Amazon Athena für Business Intelligence.....	33
4.4.4 Vergleich mit klassischen Datawarehouses (Redshift)	33
4.5 Amazon QuickSight – Interaktive Datenvisualisierung.....	33
4.6 Fazit: Vorteile der AWS-Dienste für dieses BI-Szenario	34
5 Datenmodellierung und Implementierung	34
5.1 Erstellung des Datenmodells (Fakten- und Dimensionstabellen)	34
5.2 QuellSystem-Datenmodell	35
5.3 Zielsystem-Datenmodell (Sternschema)	36
5.4 Aufbau des Sternschemas für das BI-Szenario	38
5.5 Implementierung und Optimierung des Datenmodells.....	40
6 Technische Umsetzung mit AWS-Diensten	40
6.1 Datenspeicherung in Amazon S3	42
6.1.1 Erstellung des S3 Buckets	42
6.1.2 Organisation der Rohdaten	43
6.1.3 Ergebnisse	45
6.2 ETL- Prozesse mit AWS Glue	45
6.2.1 Einrichtung des Glue-Crawlers	45
6.2.2 Durchführung des ETL-Jobs	55
6.3 Datenanalyse in Amazon Athena	63
6.3.1 Einrichtung von Amazon Athena	63
6.3.2 Datenimport.....	65
6.3.3 Berechnung der KPIs	66

<i>6.4 Datenvizualisierung mit Amazon QuickSight</i>	74
<i>6.4.1 Datenvorbereitung</i>	74
<i>6.4.2 Visualisierung der Key Performance Indikators (KPIs)</i>	75
7. Ergebnisse und Analyse	82
<i>7.1 Analyse der Geschäftsprozesse und KPIs</i>	82
<i>7.2 Bewertung der BI-Ergebnisse für die Entscheidungsfindung</i>	83
<i>7.3 Vergleich der Ergebnisse vor und nach Implementierung des BI-Systems</i>	83
<i>7.4 Diskussion über die Skalierbarkeit und Effizienz der verwendeten AWS-Dienste</i> ..	83
8. Fazit	84
<i>8.1 Zusammenfassung der Arbeit und der wichtigsten Ergebnisse</i>	84
<i>8.2 Ausblick auf mögliche Erweiterungen oder Implementierungen im realen Unternehmenskontext</i>	84
<i>8.3 Reflexion über die Rolle von Cloud-Services für die BI-Entwicklung im E-Commerce</i>	84
Abbildungsverzeichnis	85
Tabellenverzeichnis:	87
Literatur	88
KI- Verzeichnis	90
Anhang:	91

Abkürzungsverzeichnis:

Abkürzung	Bedeutung
AWS	Amazon Web Services
BI	Business Intelligence
CLV	Customer Lifetime Value
CSV	Comma-Separated Values
ETL	Extract, Transform, Load
IAM	Identity and Access Management
KPI	Key Performance Indicator
ML	Maschine Learning
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
S3	Simple Storage Service
SQL	Structured Query Language

1. Einleitung

1.1 Hintergrund und Motivation

Im digitalen Zeitalter ist **Business Intelligence (BI)** essenziell, um datenbasierte Entscheidungen zu treffen und Geschäftsstrategien zu optimieren. Unternehmen können große Datenmengen analysieren und wertvolle Erkenntnisse gewinnen [1, S. 35]. Besonders im **Fashion E-Commerce**, einem wettbewerbsintensiven Markt, sind präzise Analysen notwendig, um Markttrends frühzeitig zu erkennen und betriebliche Abläufe effizient zu steuern.

Der Modehandel im E-Commerce ist charakterisiert durch **kurze Produktlebenszyklen, hohe Retourenquoten und stark unterschiedliche Kundenpräferenzen**. Unternehmen stehen vor dem Auf, umfangreiche Mengen an Verkaufs-, Kunden- und Lieferdaten effizient zu analysieren. Traditionelle BI-Systeme stoßen dabei an **Skalierungs- und Echtzeitgrenzen**. Der Modehandel im E-Commerce ist geprägt durch kurze Produktlebenszyklen, hohe Retourenquoten und stark variierende Kundenpräferenzen. Unternehmen stehen vor der Herausforderung, große Mengen an Verkaufs-, Kunden- und Lieferdaten effizient zu analysieren. Traditionelle BI-Systeme stoßen dabei an Skalierungs- und Echtzeitgrenzen.

Cloud-basierte BI-Architekturen bieten hier eine leistungsfähige Alternative. Insbesondere **Amazon Web Services (AWS)** stellt mit skalierbaren Speicher- und Analyseplattformen eine effiziente Datenverarbeitung sicher. Durch den Einsatz moderner Cloud-Technologien können Unternehmen nicht nur **ihre Wettbewerbsfähigkeit steigern, sondern auch Kosten senken und flexibel auf Marktveränderungen reagieren**. AWS bietet hierfür eine leistungsstarke Infrastruktur mit optimierten Komponenten für datenintensive E-Commerce-Anwendungen, die eine schnelle und flexible Analyse ermöglichen.[2]

1.2 Problemstellung

Der Modehandel ist einer der größten und dynamischsten Sektoren im E-Commerce, der durch kontinuierliche Veränderungen und ein rapides Wachstum gekennzeichnet ist. Modetrends unterliegen einem ständigen Wandel, das Kundenverhalten variiert signifikant zwischen verschiedenen Märkten und die Retourenquote ist als außerordentlich hoch zu bewerten. Diese Faktoren erschweren datenbasierte Entscheidungsprozesse für Unternehmen und erfordern eine leistungsfähige sowie flexible Analyseplattform.

Im Fashion-E-Commerce stellen sich zentrale Herausforderungen:

- Dynamische Marktveränderungen: Modetrends unterliegen einem schnellen Wandel, häufig innerhalb weniger Wochen, was die effiziente Analyse großer Datenmengen erfordert.
- Hohe Retourenquoten: Im Online-Modehandel werden zwischen 30 % und 50 % der Bestellungen zurückgesendet, was erhebliche Kosten verursacht.
- Schwierige Kundenbindung: Die langfristige Bindung von Kunden gestaltet sich herausfordernd und erfordert zielgerichtete Marketingstrategien sowie personalisierte Ansätze, die mit hohen Investitionen verbunden sind.

Traditionelle E-Commerce-Analysen beschränken sich oftmals auf einfache Berichte, die lediglich vergangene Verkaufszahlen darstellen. Diese liefern jedoch keine tiefgehenden Erkenntnisse über das Kundenverhalten, Retourenmuster oder die Wirksamkeit von Marketingkampagnen. Zudem können wachsende Datenmengen bei klassischen On-Premise-BI-Lösungen zu Skalierungsproblemen führen. Cloud-Computing stellt eine skalierbare Alternative dar, um datenintensive Anwendungen effizient zu betreiben. Insbesondere Amazon Web Services (AWS) bietet eine leistungsfähige Infrastruktur, die eine umfassende Datenintegration, Echtzeitanalyse und Visualisierung ermöglicht.

1.3 Zielsetzung der Arbeit

Das Ziel dieser Arbeit besteht in der Entwicklung und Implementierung eines cloudbasierten Business-Intelligence-Systems (BI-Systems), das auf Amazon Web Services (AWS) basiert und spezifisch auf die Anforderungen einer fiktiven Fashion-E-Commerce-Plattform zugeschnitten ist. Die übergeordnete Zielsetzung ist die Konzeption einer skalierbaren Datenarchitektur, die eine umfassende Analyse von Verkaufs-, Retouren- und Kundenverhaltensdaten ermöglicht.

Ein zentraler Bestandteil der Arbeit ist die Entwicklung eines **Sternschemas** zur effizienten Speicherung und Verarbeitung der relevanten Daten. Dieses Schema bildet die Grundlage für analytische Abfragen mit **Amazon Athena**. Die Automatisierung der ETL-Prozesse (Extract, Transform, Load) erfolgt durch **AWS Glue**, um Daten aus heterogenen Quellen zu extrahieren, zu transformieren und für weiterführende Analysen bereitzustellen. Die resultierenden Erkenntnisse werden mithilfe von **Amazon QuickSight** visualisiert, das interaktive Dashboards zur Unterstützung datenbasierter Entscheidungsprozesse bereitstellt.

Spezifische Zielsetzungen der Arbeit

1. Entwicklung eines skalierbaren Datenmodells für den Fashion-E-Commerce:

- Implementierung eines BI-Systems zur effizienten Verarbeitung großer Mengen an Verkaufs-, Retouren- und Kundendaten.
- Nutzung eines Sternschemas zur Optimierung analytischer Abfragen.

- Speicherung und Verwaltung der Daten in **Amazon S3**, um eine maximale Skalierbarkeit sicherzustellen.

2. Automatisierung von ETL-Prozessen mit AWS Glue:

- Entwicklung von **ETL-Pipelines** zur Datenbereinigung, -transformation und -integration.
- Implementierung eines inkrementellen Ladevorgangs anstelle vollständiger Neuladungen zur Optimierung der Systemperformance.

3. Berechnung und Analyse zentraler betriebswirtschaftlicher Kennzahlen (KPIs):

- Analyse des **Umsatzes** nach Produktkategorien und geographischen Regionen.
- Berechnung der **Retourenquote** in Abhängigkeit von Produkttypen und Kundensegmenten.
- Bestimmung der **Konversionsrate** sowie des **Customer Lifetime Value (CLV)** zur Bewertung der Kundenbindung und des langfristigen Kundenwerts.

4. Erstellung eines interaktiven Dashboards mit Amazon QuickSight:

- Entwicklung einer benutzerfreundlichen, interaktiven **Dashboard-Umgebung** zur Visualisierung der zentralen KPIs.
- Bereitstellung **Echtzeit-basierter Analysen** zur Unterstützung datengetriebener Geschäftsentscheidungen.

Wissenschaftlicher Beitrag der Arbeit

Diese Arbeit leistet einen Beitrag zur Untersuchung der Machbarkeit und Leistungsfähigkeit eines **AWS-basierten Business-Intelligence-Systems** im Fashion-E-Commerce. Sie analysiert insbesondere, inwiefern **cloudbasierte BI-Architekturen** traditionellen **On-Premise-Systemen** hinsichtlich Skalierbarkeit, Effizienz und Flexibilität überlegen sind. Darüber hinaus wird untersucht, welche Potenziale eine **datengetriebene Unternehmenssteuerung** im E-Commerce bietet.

1.4 Aufbau der Thesis

Kapital	Inhalt
1	Einleitung: Einführung in das Thema, Zielsetzung, Aufbau
2	Grundlagen von BI: Definitionen, Relevanz im E-Commerce
3	Geschäftsprozesse: Abläufe und relevante KPIs
4	Technische Umsetzung: Nutzung von AWS-Diensten

5	Datenmodellierung: Sternschema und ETL-Prozess
6	Analyse der Ergebnisse: Effizienz und Optimierungen
7	Fazit und Ausblick: Zusammenfassung und Empfehlungen

Tabelle 1: Aufbau der Thesis – Kapitelübersicht

2. Theoretischer Hintergrund

Der theoretische Rahmen dieser Arbeit bildet die Grundlage für das Verständnis der zentralen Konzepte und liefert die Basis für die nachfolgenden Analysen und Implementierungen. Um die Rolle von Business Intelligence (**BI**) im E-Commerce, insbesondere im Fashion-Sektor, fundiert zu erfassen, ist es essenziell, die grundlegenden Ziele und Methoden von BI detailliert zu untersuchen.

Ein weiterer bedeutender Aspekt ist die Nutzung moderner Cloud-Technologien, die eine zentrale Rolle bei der Implementierung effizienter BI-Lösungen spielen. Leistungsfähige Dienste wie jene von Amazon Web Services (AWS) eröffnen neue Möglichkeiten zur Verarbeitung und Analyse großer Datenmengen, wodurch innovative Ansätze für datengetriebene Entscheidungsprozesse ermöglicht werden.

Dieses Kapitel beginnt mit einer Einführung in die theoretischen Grundlagen von Business Intelligence. Anschließend wird die spezifische Bedeutung von BI für den Fashion-E-Commerce näher betrachtet. Darüber hinaus werden die relevanten Cloud-Technologien beschrieben, die moderne BI-Systeme unterstützen. Abschließend erfolgt eine detaillierte Erläuterung der AWS-Dienste, die im Rahmen dieser Arbeit zur Entwicklung eines skalierbaren und leistungsfähigen BI-Systems genutzt werden.

2.1 Grundlagen von Business Intelligence (BI)

2.1.1 Definition:

„Business Intelligence bezeichnet die faktenbasierte Entscheidungsunterstützung auf der Grundlage einer strukturierten Datenerhebung und Analyse. Intelligence kann in diesem Kontext auch als zu gewinnendes 'Verständnis' für die Wirkzusammenhänge in der Organisation bezeichnet werden“ [1, S. 36].

Lohmanns Definition unterstreicht die zentrale Rolle von Business Intelligence (BI) als datengetriebenes Instrument zur Entscheidungsunterstützung. BI ermöglicht die Sammlung und systematische Analyse von Daten aus unterschiedlichen Quellen - wie etwa Verkaufszahlen, Kundenbewertungen oder Lagerbestände. Durch das strukturierte Sammeln und Analysieren dieser Daten können Unternehmen nicht nur operative Prozesse optimieren, sondern auch fundierte strategische Entscheidungen treffen. BI kann beispielsweise helfen, Trends im Kundenverhalten frühzeitig zu erkennen, so dass Unternehmen ihr Angebot entsprechend anpassen können. In einem

dynamischen Geschäftsumfeld, in dem es auf schnelle Anpassungsfähigkeit und Datenverfügbarkeit ankommt, ist BI eine unverzichtbare Grundlage für langfristige Wettbewerbsfähigkeit.

Eine weitere Definition von Business Intelligence unterscheidet zwischen einem engen und einem weiten Verständnis, wie in Abbildung 1 dargestellt:

Enges BI-Verständnis: Business Intelligence im engeren Sinne umfasst spezifische Kernanwendungen wie OLAP (Online Analytical Processing), Management Information Systems (MIS) und Executive Information Systems (EIS), die eine direkte Unterstützung von Entscheidungsprozessen ermöglichen. [3, S. 3]

Analyseorientiertes BI-Verständnis: Diese Perspektive erweitert das enge Verständnis um zusätzliche Anwendungen wie Text Mining, Data Mining, Ad-hoc-Reporting sowie Planungs- und Konsolidierungssysteme. Sie bietet interaktive Funktionen, die speziell auf die Bedürfnisse von Entscheidungsträgern ausgerichtet sind. [3, S. 4]

Weites BI-Verständnis: Business Intelligence im weiteren Sinne umfasst alle Anwendungen, die direkt oder indirekt zur Entscheidungsunterstützung beitragen. Dies beinhaltet u.a. die Bereitstellung, Speicherung und Analyse von Daten. [3, S. 4]

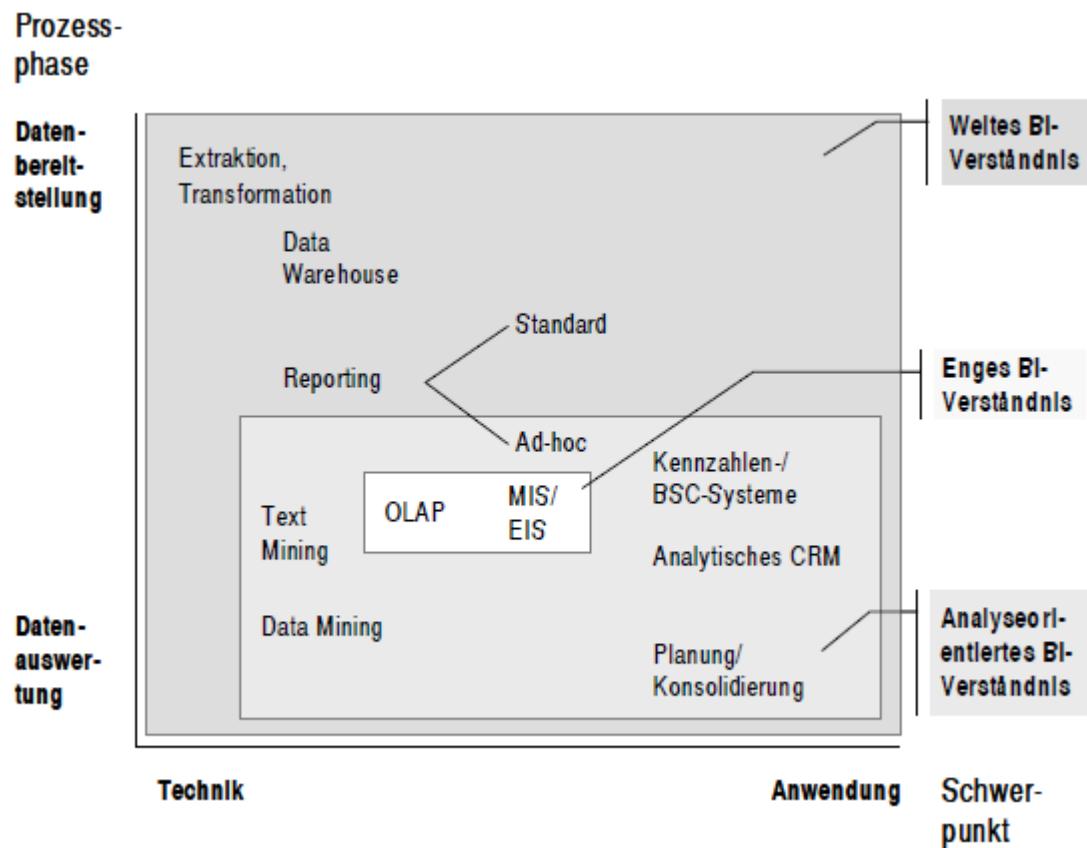


Abbildung 1: Ordnungsrahmen für BI-Definitionen

2.1.2 Hauptziele von Business Intelligence

Business Intelligence verfolgt eine Vielzahl an Zielen, die sowohl zur **Optimierung betrieblicher Prozesse** als auch zur **Unterstützung des Managements** bei der Entscheidungsfindung beitragen. Die wesentlichen Zielsetzungen lassen sich wie folgt zusammenfassen:

1. **Unterstützung der Entscheidungsfindung:** BI-Systeme sollen das Management durch die Bereitstellung relevanter Daten und Informationen in der Entscheidungsfindung unterstützen. [3, S. 5]
2. **Verbesserung der Unternehmensleistung:** Die Analyse von Daten und die Generierung wertvoller Erkenntnisse sollen zur Steigerung der Unternehmensleistung beitragen. [3, S. 261]
3. **Optimierung von Geschäftsprozessen:** BI-Systeme sollen ineffiziente Abläufe identifizieren und Potenziale zur Prozessoptimierung aufzeigen. [3, S. 258]
4. **Steigerung der Wettbewerbsfähigkeit:** Durch die Nutzung von BI können Unternehmen Markttrends frühzeitig erkennen und Wettbewerbsanalysen durchführen, um ihre Marktposition zu stärken. [3, S. 258]

5. **Effiziente Datenanalyse:** BI-Systeme ermöglichen eine schnelle und effiziente Analyse großer Datenmengen, um wertvolle Erkenntnisse zu gewinnen. [3, S. 114]
6. **Flexibilität und Anpassungsfähigkeit:** BI-Systeme sollen flexibel und skalierbar sein, um sich an verändernde Geschäftsanforderungen anzupassen. [3, S. 207]

Durch die konsequente Verfolgung dieser Ziele wird Business Intelligence zu einem zentralen Bestandteil der modernen Unternehmensführung, der sowohl strategische, taktische als auch operative Entscheidungen effektiv unterstützt.

2.1.3 Methoden

Business Intelligence (BI) umfasst eine Vielzahl von Methoden, die darauf abzielen, Daten zu erfassen, zu analysieren und in entscheidungsrelevante Informationen umzuwandeln. Die wesentlichen Methoden lassen sich wie folgt zusammenfassen:

2.1.3.1 Transformationsprozess – ETL

Der **Transformationsprozess (ETL: Extract, Transform, Load)** stellt einen essenziellen Bestandteil der **Datenintegration** in Business Intelligence dar. Er umfasst alle Aktivitäten, die erforderlich sind, um **operative Daten in entscheidungsrelevante Informationen** zu überführen. Ein zentraler Aspekt dieses Prozesses ist die Sicherstellung der **Datenqualität**, indem die Daten **gefiltert, harmonisiert, aggregiert und angereichert** werden. [4, S. 26-27]

1. **Filterung:** Die Filterung beinhaltet die **Extraktion** von Daten aus operativen Systemen und die Bereinigung syntaktischer oder inhaltlicher Fehler, bevor die Daten in das **Data Warehouse** übernommen werden. Dieser Schritt gewährleistet, dass ausschließlich **relevante und korrekte Daten** für die weitere Verarbeitung genutzt werden. [4, S. 26-27]

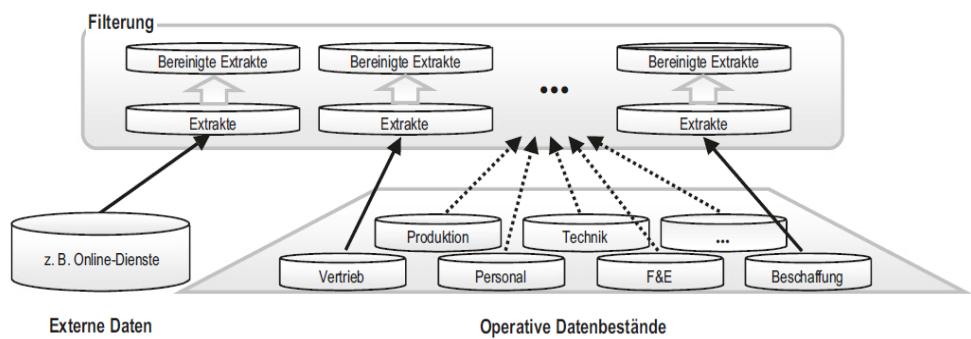


Abbildung 2: Erste Transformationsschicht Filterung [4, S. 27]

2. **Harmonisierung:** Die Harmonisierung bezieht sich auf die **Angleichung unterschiedlicher Datenformate und Strukturen**, um eine **einheitliche und konsistente Basis** für Analysen zu schaffen. [4, S. 26-27]

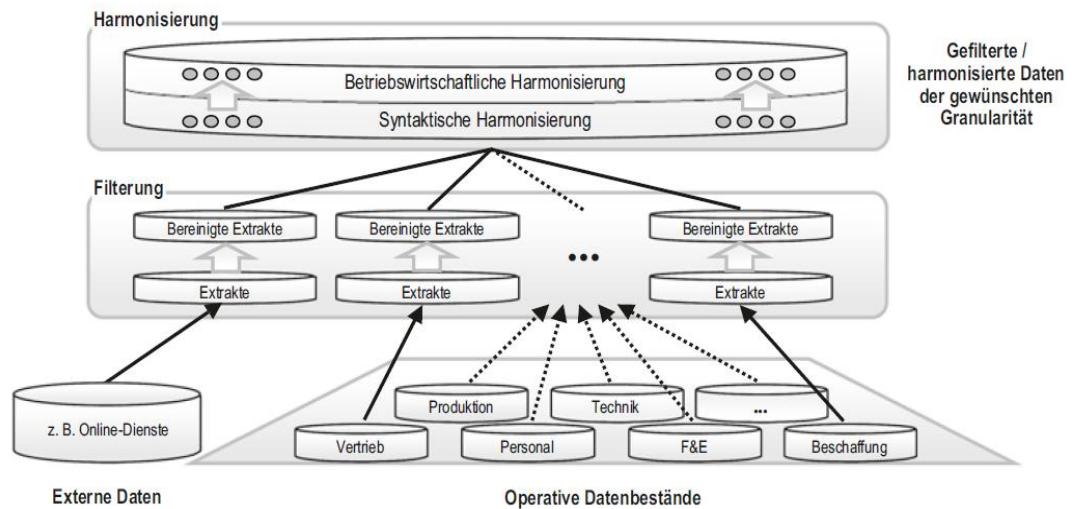


Abbildung 3: Zweite Transformationsschicht Harmonisierung [4, S. 31]

3. **Aggregation:** Aggregation bedeutet die **Zusammenfassung** von gefilterten und harmonisierten Daten, um eine **verdichtete Form** bereitzustellen. Dies erleichtert die Analyse großer Datenmengen, indem die Daten auf **wesentliche Kennzahlen** reduziert werden. [4, S. 26-27]

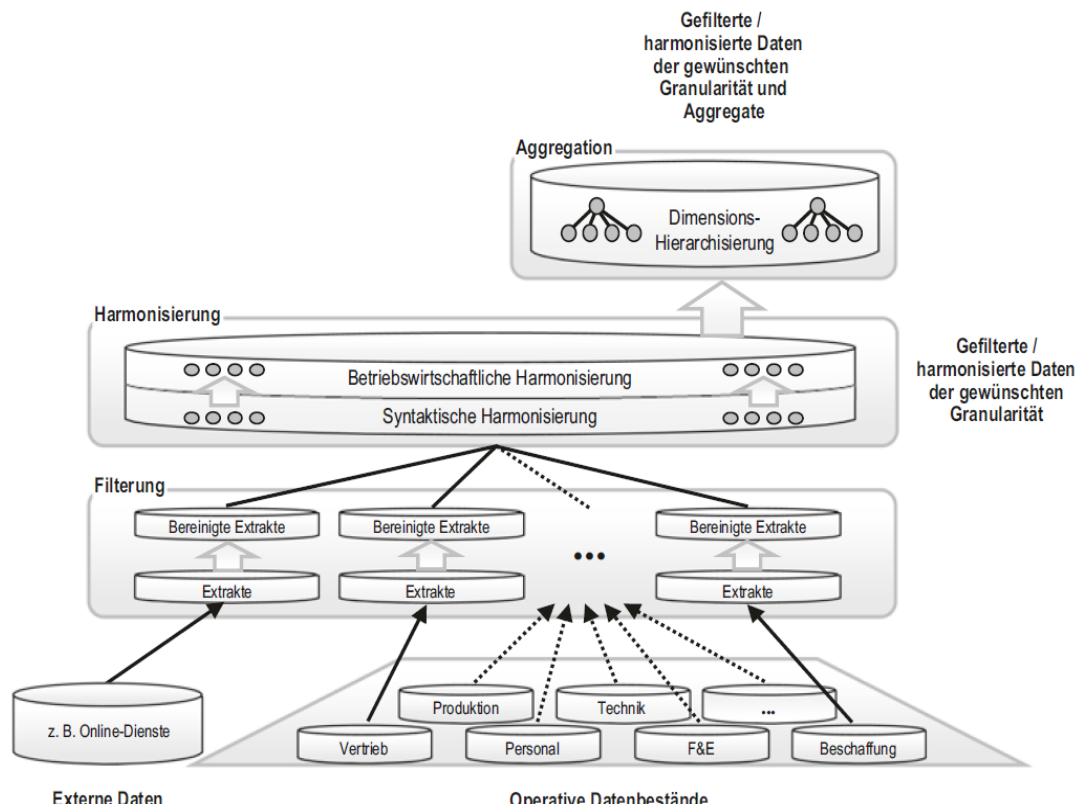


Abbildung 4: Dritte Transformationsschicht Aggregation [4, S. 34]

4. **Anreicherung:** Die Anreicherung umfasst die **Berechnung und Speicherung betriebswirtschaftlicher Kennzahlen** wie Umsatz, Gewinnmargen oder Retourenquoten, die aus den gefilterten und harmonisierten Daten abgeleitet werden. [4, S. 26-27]

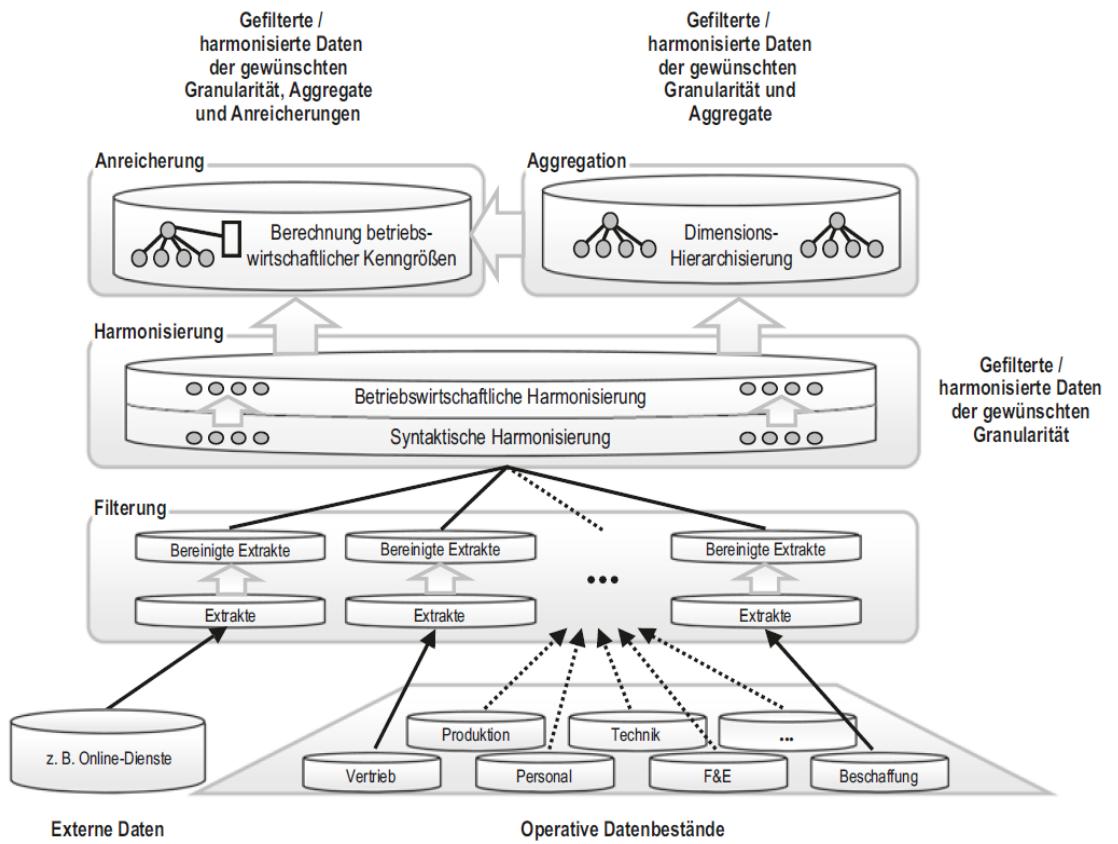


Abbildung 5: Vierte Transformationsschicht Anreicherung [4, S. 36]

2.1.3.2 Data Warehousing:

Data Warehousing umfasst Technologien und Konzepte, die darauf abzielen, **große Datenmengen systematisch zu speichern, zu organisieren und für analytische Zwecke bereitzustellen**. Ursprünglich in den 1980er Jahren entwickelt, hat sich das Konzept in den letzten Jahrzehnten erheblich weiterentwickelt und ist heute eine Schlüsselkomponente moderner BI-Architekturen. [5]

Besonders im **Fashion-E-Commerce** spielt das Data Warehousing eine bedeutende Rolle, da dieser Sektor stark auf datenbasierte Entscheidungsprozesse angewiesen ist. Beispiele für spezifische Anwendungsgebiete sind:

- **Kundensegmentierung:** Durch die Speicherung und Analyse von Kaufhistorien können Unternehmen **zielgerichtete Produktempfehlungen** erstellen.
- **Trendanalysen:** Historische Verkaufsdaten ermöglichen die **frühzeitige Identifikation von Modetrends** und eine gezielte Produktentwicklung.
- **Lager- und Bestandsmanagement:** Mithilfe von **Data Warehousing** lassen sich saisonale Nachfrageschwankungen prognostizieren und die **Bestandsplanung optimieren**, um sowohl Überbestände als auch Fehlbestände zu minimieren.

- **Retourenmanagement:** Durch die Analyse von Retourendaten können Unternehmen **Muster im Rückgabeverhalten** erkennen und gezielte Maßnahmen ergreifen, um die Retourenquote zu senken.

Das **Data Warehouse** bildet somit eine zentrale Informationsquelle, um operative Daten in strategisch relevante Erkenntnisse umzuwandeln und den Herausforderungen im dynamischen Umfeld des Fashion E-Commerce gerecht zu werden.

2.1.3.3 Datenanalyse

1- OLTP und OLAP: OLTP (On-Line Transaction Processing) und OLAP (On-Line Analytical Processing) sind zwei fundamentale Technologien in der Datenverarbeitung, die unterschiedliche Anwendungszwecke erfüllen: [6]

- **OLTP:** OLTP-Systeme sind transaktionsorientiert und dienen der Abwicklung von Kundenanfragen, wie Buchungen oder Bestellungen. Sie sind datenbankzentriert und darauf ausgelegt, Echtzeitdaten zu verarbeiten. [6]
- **OLAP:** OLAP-Systeme hingegen sind analytik_orientiert und unterstützen Wissensarbeiter bei der Datenanalyse. Sie verarbeiten große Datenmengen, ermöglichen die Aggregation von Informationen und arbeiten oft mit multidimensionalen Datenmodellen wie Stern- oder Schneeflockenschemata. [6]

2- Data Mining: Data Mining ist der Prozess, intelligente Methoden anzuwenden, um Datenmuster zu extrahieren. Es kombiniert künstliche Intelligenz, statistische Analyse und Datenbankmanagementsysteme, um Wissen aus großen Datenmengen oder einem Data Warehouse zu gewinnen. Dabei unterscheidet es sich von traditionellen Datenbankabfragen, da die Abfragen in Data Mining oft schlecht definiert und das Ergebnis nicht präzise, sondern unscharf ist. Dies ermöglicht es, neue Muster und Zusammenhänge zu entdecken, die bei herkömmlichen Datenbankabfragen verborgen bleiben. Ein Beispiel ist die Analyse von Kundenverhalten im Einzelhandel, bei der häufig gekaufte Artikelkombinationen wie „Brot und Milch“ oder „Bier und Chips“ identifiziert werden können, um die Verkaufsstrategie zu optimieren. [6]

3- Predictive Analytics: Predictive Analytics umfasst prädiktive Modellierung, statistische Techniken und Data Mining, die verwendet werden, um zukünftige Trends vorherzusagen. Zu den häufig verwendeten Modellen zählen lineare Regression, logistische Regression, Entscheidungsbäume, Random Forests und Zeitreihenprognosemodelle wie ARIMA und das Holt-Winters-Verfahren. [7, S. 11]

2.1.3.4 Datenvisualisierung und Berichterstellung

Die Rolle von Visualisierungen in Business-Intelligence-Systemen ist zentral, da visuelle Darstellungen komplexe Datenmengen auf verständliche Weise kommunizieren. *Laut Halfmann und Schüller (2020)*

ist die Verarbeitung visueller Informationen evolutionär bedingt eine der effizientesten Fähigkeiten des menschlichen Geistes. Ziel der Datenvisualisierung ist es, Informationen so aufzubereiten, dass sie Kommunikation, Informationsgewinn und Entscheidungsprozesse effektiv unterstützen. Dies erfordert eine Kombination aus Psychologie, Informatik und Design, um leicht verständliche und zugleich aussagekräftige Darstellungen zu erstellen. [8, S. 140]

In der Praxis ist es entscheidend, die jeweils geeignete Visualisierung zu wählen, um die Botschaft klar zu transportieren. Unternehmen setzen zunehmend auf spezialisierte Visualisierungswerkzeuge wie Tableau, Power BI oder AWS QuickSight, um komplexe Daten in Dashboards und interaktiven Berichten aufzubereiten. Diese Visualisierungen ermöglichen es Entscheidungsträgern, fundierte Entscheidungen zu treffen und Wettbewerbsvorteile zu generieren. Erfolgreiche BI-Visualisierungen erfordern jedoch nicht nur technische Tools, sondern auch unternehmerisches Wissen, um den größtmöglichen Nutzen zu erzielen [8, S. 140]

2.2 Bedeutung von BI für den E-Commerce-Sektor, speziell im Fashion-Bereich

Business Intelligence (BI) hat im E-Commerce eine zentrale Bedeutung erlangt, da datenbasierte Entscheidungsprozesse Unternehmen entscheidende Wettbewerbsvorteile verschaffen und die Anpassung an sich schnell ändernde Marktbedingungen ermöglichen. Insbesondere im **Fashion-Sektor**, der durch dynamische Modetrends und individuelle Kundenpräferenzen gekennzeichnet ist, spielt BI eine essenzielle Rolle.

Herausforderungen im Fashion-E-Commerce

1. Schnelle Modetrends:

- Die Modebranche ist durch eine hohe Dynamik und kurzlebige Trends geprägt, wodurch eine schnelle Anpassung des Produktangebots erforderlich ist.
- Laut Pan et al. (2021) müssen "E-Commerce-Unternehmen den Anforderungen an Personalisierung, Komfort und Interaktivität gerecht werden, um die Kundenzufriedenheit zu verbessern". [9, S. 26]
- BI ermöglicht es, Trends frühzeitig durch die Analyse von Verkaufsdaten und Social-Media-Aktivitäten zu identifizieren und darauf strategisch zu reagieren.

2. Retourenmanagement:

- Hohe Rücksendequoten stellen eine erhebliche finanzielle Belastung für E-Commerce-Unternehmen dar.

- Pan et al. (2021) betonen, dass intelligente Technologien dabei helfen, "Transaktionskosten zu senken, die Kaufkraft zu verbessern und den Prozess zu integrieren". [9, S. 26]
- BI-gestützte Analysen helfen, Muster in Retouren zu identifizieren und Strategien zur Reduktion der Rücksendungen zu entwickeln.

3. Lager- und Bestandsmanagement:

- Eine effiziente Lagerverwaltung ist entscheidend, um Überbestände oder Engpässe zu vermeiden.
- Laut Pan et al. (2021) wird die "tiefgehende Anwendung digitaler Technologien zunehmend beschleunigt". [9, S. 26]
- BI kann Bestände präziser verwalten und durch datengestützte Entscheidungsprozesse ineffiziente Abläufe optimieren.

4. Verändertes Kundenverhalten und Marktmechanismen

- Die Digitalisierung hat das Kaufverhalten grundlegend verändert, wodurch traditionelle Einzelhändler zunehmend den direkten Kontakt zu ihren Kunden verlieren.
- Laut Pan et al. (2021) beginnt die Digitalisierung, "den institutionellen Einzelhandel als primäre Schnittstelle zu den Kunden zu verdrängen". [9, S. 26]
- BI ermöglicht eine präzise Analyse des Kaufverhaltens und die Entwicklung datenbasierter Marketingstrategien, um eine langfristige Kundenbindung zu stärken.

BI- und E-Commerce-Architektur

Die Integration von Business Intelligence und E-Commerce ermöglicht Unternehmen, datenbasierte Entscheidungen effizient zu treffen und ihre Wettbewerbsfähigkeit zu steigern. Abbildung 2 veranschaulicht die Architektur, wie BI im E-Commerce eingesetzt wird, um Prozesse zu optimieren und Wettbewerbsvorteile zu erzielen. [10]

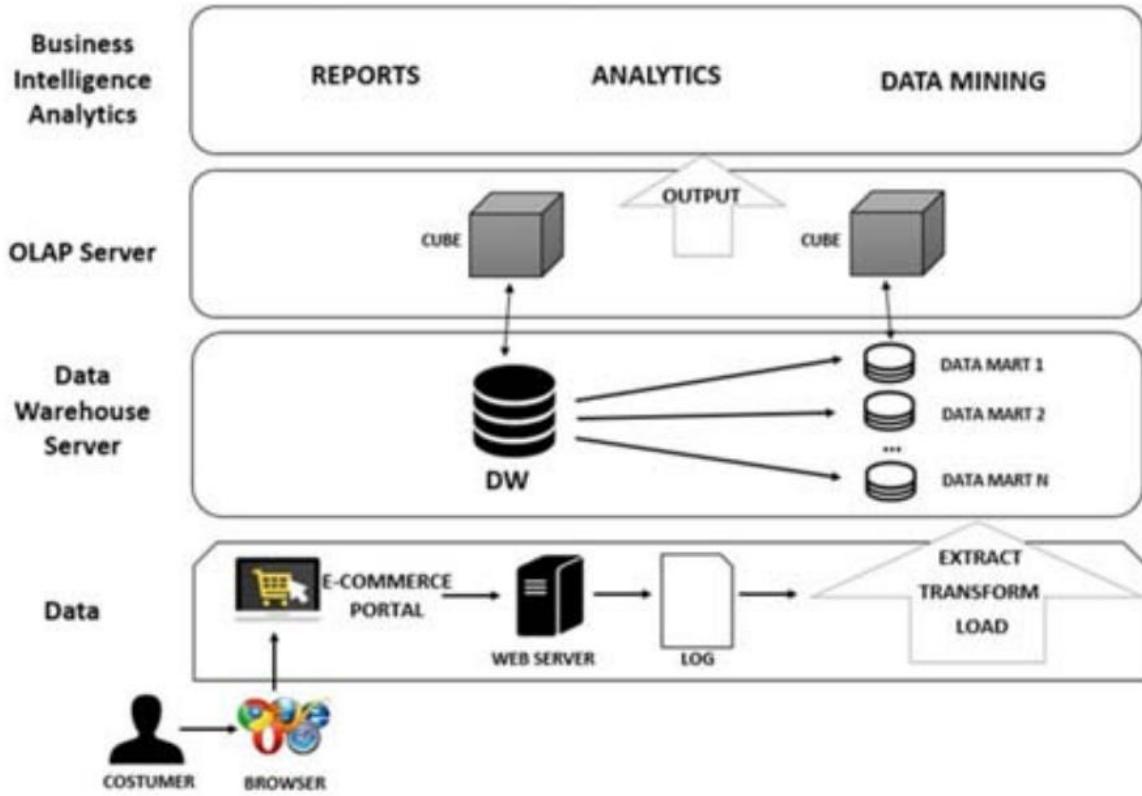


Abbildung 6: Business-Intelligence- und E-Commerce-Architektur [9]

Diese Architektur zeigt, wie Daten von Kundenportalen und Webservern in ein Data Warehouse eingespeist, über OLAP-Systeme verarbeitet und schließlich für Analysen, Berichte und Data Mining bereitgestellt werden. Diese Integration ermöglicht es Unternehmen, Kundenverhalten zu analysieren, Lagerbestände zu verwalten, Markttrends zu identifizieren und gezielte Marketingkampagnen durchzuführen.

2.3 Überblick über Cloud-Technologien und AWS-Dienste im BI-Kontext

Moderne BI-Systeme nutzen zunehmend Cloud-Technologien, um Flexibilität, Skalierbarkeit und Kosteneffizienz zu gewährleisten. Amazon Web Services (AWS) bietet eine Vielzahl von Diensten, die speziell für BI-Anwendungen entwickelt wurden.

Amazon Web Services: Gemäß der [offiziellen AWS-Website](#) ist **Amazon Web Services (AWS)** die weltweit führende **Cloud-Plattform**, die über **200 voll funktionsfähige Dienste** aus **Rechenzentren auf der ganzen Welt** bereitstellt. AWS ermöglicht es Unternehmen, **schnell skalierbare, sichere und hochverfügbare Cloud-Lösungen** für eine Vielzahl von Anwendungsfällen zu implementieren. [11] (Was ist AWS?)

Elastic Computing Cloud EC2: Amazon EC2 ist ein skalierbarer Cloud-Computing-Dienst, der Unternehmen ermöglicht, virtuelle Server (Instanzen) bedarfsgerecht bereitzustellen und zu verwalten. Diese Server können individuell hinsichtlich Rechenleistung, Speicher und Netzwerkkapazität angepasst werden, um flexible und kosteneffiziente IT-Ressourcen für BI-Anwendungen bereitzustellen.[11]. (Amazon Elasting Computing Cloud).

Identity and Access Management IAM ist AWS IAM ist ein Webservice zur Zugriffskontrolle, mit dem sich Benutzerrechte innerhalb der AWS-Umgebung verwalten lassen. Mithilfe von IAM-Richtlinien kann genau definiert werden, welche Nutzer und Rollen Zugriff auf bestimmte AWS-Ressourcen erhalten. Dadurch werden Authentifizierung und Autorisierung zentral gesteuert, um Datensicherheit und Compliance-Anforderungen zu gewährleisten. [11] (was ist IAM).

Amazon Simple Storage Service (S3) ist ein Cloud-basierter Objektspeicherdiensst, der eine sichere, skalierbare und hochverfügbare Speicherung großer Datenmengen ermöglicht. Unternehmen nutzen S3 insbesondere für:

- Data Lakes
- Backup- und Archivierungslösungen
- Big-Data-Analysen
- IoT-Anwendungen
- Web- und mobile Applikationen

Der Dienst bietet umfangreiche Datenmanagement- und Sicherheitsfunktionen, darunter Zugriffssteuerungen und Compliance-Mechanismen, um unternehmenskritische Informationen sicher zu verwalten.[11] (was ist S3?).

AWS Glue ist ein serverloser Dienst zur Datenintegration, der Unternehmen ermöglicht, Daten aus heterogenen Quellen zu extrahieren, zu transformieren und in Data Warehouses oder Data Lakes zu laden. Mithilfe von AWS Glue lassen sich ETL-Pipelines (Extract, Transform, Load) einfach konfigurieren und in andere AWS-Services wie Amazon S3, Athena und Redshift integrieren.

Die wichtigsten Funktionen von AWS Glue:

- Automatische Katalogisierung von Datenquellen
- Visuelle Erstellung von ETL-Jobs
- Serverlose Architektur – keine Infrastrukturverwaltung erforderlich

Dadurch eignet sich der Dienst ideal für Big-Data-Analysen, maschinelles Lernen und datengetriebene Anwendungen. [11] (was ist AWS Glue).

Amazon Redshift ist ein vollständig verwaltetes, skalierbares Cloud-Data-Warehouse, das speziell für analytische Abfragen und datenintensive Workloads entwickelt wurde. Durch die spaltenorientierte Speicherung und parallele Verarbeitung können große Datenmengen effizient analysiert werden.

Wichtige Merkmale von Amazon Redshift:

- Optimiert für Business Intelligence und Reporting
- Nahtlose Integration mit AWS S3 und AWS Glue
- Unterstützung für komplexe SQL-Abfragen auf Petabyte-großen Datenmengen

Damit eignet sich Redshift ideal für Unternehmen, die leistungsfähige BI-Lösungen mit hoher Performance benötigen. [11](was ist AWS Redshift)

AWS Quicksight ist ein **BI-Service für interaktive Datenvisualisierung und Dashboards**. Er ermöglicht es Unternehmen, **Daten aus verschiedenen Quellen zu aggregieren, zu analysieren und grafisch darzustellen**.

Vorteile von **AWS QuickSight**:

- **Benutzerfreundliche, interaktive Dashboards**
- **Automatisierte Erkenntnisse durch maschinelles Lernen**
- **Einfache Integration mit AWS-Datenquellen und Drittanbieter-Tools**

Durch seine **skalierbare Cloud-Architektur** kann QuickSight für **individuelle Nutzer oder große Unternehmensteams** eingesetzt werden. [11] (was ist Quicksight?)

Amazon CloudWatch ist ein Überwachungs- und Analyse-Service, der AWS-Ressourcen und Anwendungen in Echtzeit analysiert.

Hauptfunktionen von CloudWatch:

- Erfassung von Systemmetriken wie CPU-Auslastung, Speicherverbrauch und Netzwerkauslastung
- Erstellung benutzerdefinierter Dashboards zur Überwachung von Cloud-Ressourcen
- Automatische Alarne und Benachrichtigungen bei Erreichen kritischer Schwellenwerte

CloudWatch trägt dazu bei, **Betriebskosten zu optimieren, Ressourcen effizient zu nutzen und potenzielle Systemprobleme frühzeitig zu erkennen**. [11] (was ist CloudWatch?).

3 Szenario-Beschreibung und Geschäftsprozesse

3.1 Beschreibung der fiktiven Fashion-E-Commerce-Plattform

Die in dieser Arbeit untersuchte fiktive Fashion-E-Commerce-Plattform dient als Grundlage zur Implementierung und Analyse von Business-Intelligence-Anwendungen. Ziel ist es, eine realistische Plattform zu simulieren, die typische Funktionen eines modernen Online-Shops abbildet. Dadurch wird eine praxisnahe Umgebung geschaffen, die die Analyse und Optimierung von Geschäftsprozessen ermöglicht. Die Plattform zeichnet sich durch folgende Schlüsselfunktionen aus:

Produktpräsentation: Eine Vielzahl von Produkten aus Kategorien wie Kleidung, Schuhe und Accessoires wird angeboten. Zu jedem Produkt stehen detaillierte Informationen wie Preis, Marke, Verfügbarkeit und Rabatte bereit.

Warenkorbsystem: Kunden können Produkte in einen virtuellen Warenkorb legen, diesen bearbeiten und den Kaufprozess abschließen.

Zahlungsintegration: Es werden diverse Zahlungsmethoden unterstützt, darunter Kreditkarte, PayPal, Banküberweisung und Nachnahme.

Retourenmanagement: Kunden haben die Möglichkeit, Produkte zurückzusenden und die Rückgründe anzugeben. Diese Daten werden für spätere Analysen genutzt.

Marketingintegration: Die Plattform ermöglicht Kampagnen wie "Black Friday", Sommerveräußerungen und personalisierte Rabatte, um die Verkaufszahlen zu steigern.

Mehrsprachigkeit und regionale Anpassung: Die Plattform unterstützt verschiedene Sprachen und Währungen, um unterschiedliche geografische Märkte abzudecken.

Diese Funktionen bilden eine realistische Grundlage, um typische Herausforderungen und Optimierungspotenziale im Fashion-E-Commerce zu analysieren.

3.2 Detaillierte Darstellung der Geschäftsprozesse

3.2.1 Bestellprozess

- Kunden wählen Produkte aus und legen diese in den Warenkorb.
- Nach Abschluss des Einkaufs durchlaufen sie den Bezahlprozess, bei dem sie die bevorzugte Zahlungsmethode auswählen.
- Die Bestellung wird im System gespeichert und eine Bestellbestätigung wird generiert.

3.2.2 Zahlungsprozess

- Die Plattform integriert sichere Zahlungsgateways wie PayPal und Stripe.

- Transaktionen werden in Echtzeit überprüft, um betrügerische Aktivitäten zu vermeiden.
- Der Zahlungsstatus („erfolgreich“, „fehlerhaft“) wird gespeichert und mit der Bestellung verknüpft.

3.2.3 Lieferprozess

- Nach der Bestellbestätigung erfolgt die automatische Erstellung eines Lieferauftrags.
- Kunden können den Lieferstatus über ein Tracking-System einsehen.
- Typische Statusmeldungen umfassen „versandt“, „zugestellt“ und „rückgesendet“.

3.2.4 Retourenmanagement

- Kunden haben die Möglichkeit, Produkte zurückzusenden und die Gründe für die Rücksendung anzugeben (z. B. „falsche Größe“, „beschädigtes Produkt“).
- Das System speichert die Rücksendedaten sowie die Erstattungsbeträge.
- Diese Daten werden genutzt, um das Produktangebot und die Logistik zu optimieren.

3.2.5 Marketingkampagnen

- Es werden Kampagnen wie "Black Friday", "Sommer-Sales" oder individuelle Rabatte durchgeführt.
- Ziel ist es, Kunden zu aktivieren und die Verkaufszahlen in spezifischen Zeiträumen zu steigern.
- Erfolgsmetriken wie Umsatzsteigerung oder Konversionsraten werden analysiert, um die Effektivität der Kampagnen zu bewerten.

3.3 Identifikation und Definition der relevanten KPI

Die Performance der Plattform wird anhand von **Key Performance Indikators (KPIs)** gemessen. Die folgenden KPIs wurden definiert und analysiert:

1. Customer Lifetime Value (CLV) nach Produktkategorie

Definition:

- Der **CLV (Customer Lifetime Value)** gibt an, wie viel ein Kunde durchschnittlich über die gesamte Geschäftsbeziehung hinweg ausgibt.
- Die Analyse erfolgt **auf Basis der Produktkategorie**, um die **wertvollsten Produktsegmente** zu identifizieren.

Berechnung:

- **CLV = durchschnittlicher Umsatz pro Kunde über mehrere Bestellungen**
- Vergleich zwischen **Premium- und Nicht-Premium-Kunden**
- Durchschnittlicher Bestellwert pro Kunde

Visualisierung:

- **Vergleichsdiagramm** für CLV zwischen Produktkategorien
- **Cluster-Analyse**, um Kaufverhalten und CLV zu korrelieren

2. Retourenquote nach Produktkategorie und Kundensegment

Definition:

- Zeigt den Anteil der retournierten Produkte im Verhältnis zu den Gesamtbestellungen.
- Unterscheidung zwischen **Premium- und Nicht-Premium-Kunden**.

Berechnung:

- **Retourenquote = (Anzahl retournierten Bestellungen / Gesamtbestellungen) × 100**

Visualisierung:

- **Balkendiagramm:** Retourenquote pro Kategorie
- **Heatmap:** Retourenquote nach Monat

3. Marktanteil pro Kategorie basierend auf Umsatz und Retouren

Definition:

- Zeigt die **Verteilung des Umsatzes** pro Produktkategorie.
- Berücksichtigt den **Einfluss von Retouren auf die Netto-Umsätze**.

Berechnung:

- **Marktanteil (%) = (Umsatz einer Kategorie / Gesamtumsatz) × 100**
- **Berücksichtigung von Rücksendungen, um Netto-Umsätze korrekt zu berechnen.**

Visualisierung:

- **Kreisdiagramm (Marktanteile der Kategorien am Gesamtumsatz)**

- **Balkendiagramm (Verhältnis von Umsatz zu Retouren pro Kategorie)**

4. Umsatz pro Bestellung und Kundenbindung nach Land

Definition:

- Analyse der **durchschnittlichen Bestellwerte pro Land**, um lukrative Märkte zu identifizieren.
- Berechnung der **Kundenbindung durch Anzahl der Bestellungen pro Kunde**.

Berechnung:

- **Durchschnittlicher Bestellwert = Gesamtumsatz eines Landes / Anzahl der Bestellungen**
- **Kundenbindung = durchschnittliche Anzahl der Bestellungen pro Kunde**

Visualisierung:

- **Balkendiagramm:** Durchschnittlicher Bestellwert nach Land
- **Heatmap:** Umsatzverteilung über verschiedene Regionen

5. Monatliche Bestellwachstumsrate mit Moving Average (gleitender Durchschnitt)

Definition:

- Verfolgt das monatliche Wachstum der Bestellungen und die Umsatzentwicklung.

Berechnung:

- **Wachstumsrate (%) = [(Umsatz aktueller Monat – Umsatz Vormonat) / Umsatz Vormonat] × 100**
- **Gleitender Durchschnitt über mehrere Monate zur Trendanalyse**

Visualisierung:

- **Liniendiagramm:** Umsatzentwicklung über die Monate
- **Trendanalyse durch Moving Average zur Erkennung von Wachstumsphasen**

3.4 Generierung der Rohdaten für die Plattform

Die Qualität der zugrunde liegenden Daten spielt eine entscheidende Rolle für die Aussagekraft und Zuverlässigkeit von Business-Intelligence-Analysen. Im Rahmen dieser Arbeit wurde ein **synthetischer Datensatz** erstellt, der eine realistische Nachbildung von Geschäftsprozessen im Fashion-E-Commerce ermöglicht. Dieser Datensatz wurde mithilfe von **Python und Pandas** generiert und umfasst verschiedene Dimensionstabellen (Kunden, Produkte) sowie eine Faktentabelle (Bestellungen), die die Transaktionsdaten modelliert.

3.4.1 Methodik der Datengenerierung

Um eine fundierte Datenbasis für die Business-Intelligence-Analysen zu schaffen, wurden die folgenden Datenstrukturen entwickelt:

Kundendaten (Users-Table): Enthält simulierte demografische Merkmale (Alter, Geschlecht, Land, Premium-Mitgliedschaft). Die Verteilung der Kundenattribute basiert auf **realistischen Marktanteilen** und wurde mit zufallsbasierten Algorithmen modelliert.

Produktdaten (Products Table): Enthält eine Auswahl von Bekleidungs-, Schuh- und Accessoire-Produkten. Die Preisspannen und Kategorien wurden an bestehende Marktstrukturen angelehnt, um praxisnahe Analysen zu ermöglichen.

Bestelldaten (Orders Table): Enthält simulierte Kaufvorgänge, wobei Faktoren wie Saisonabhängigkeit, Retourenverhalten und Warenkorbgröße berücksichtigt wurden.

Die Daten wurden so generiert, dass sie **realistische Muster widerspiegeln**, wie sie in einem typischen Online-Modehandel auftreten.

Die Daten wurden mit **Python und Pandas** erzeugt und als CSV-Dateien gespeichert. Der Code befindet sich in **den Anhängen 1, 2 und 3**.

3.4.2 Generierung der Kundendaten (Users Table)

Die Users-Table enthält die wichtigsten demografischen und geografischen Informationen der Kunden. Es wurde darauf geachtet, eine globale Kundenbasis mit verschiedenen Ländern und Altersgruppen zu simulieren. Zusätzlich wurde eine realistische Premium-Mitgliedschaftsrate von 30 % integriert.

Datenfelder der Users-Tabelle:

Attribute	Beschreibung
User_ID	Eindeutige ID des Kunden
First_Name	Zufällig generierter Vorname

Last_Name	Zufällig generierter Vorname
Gender	Männlich, Weiblich oder Non-Binary
Age	Zufälliges Alter zwischen 18-75 Jahren
Country	Herkunftsland des Kunden
E-Mail	Generierte E-Mail-Adresse mit realistischen Domains
Signup_Date	Datum der Registrierung (2018-2024)
Total_Spent	Gesamtausgaben des Kunden im Shop
Premium_Member	Gibt an, ob der Kunde ein Premium-Mitglied ist

Tabelle 2: Datenfelder der Users-Tabelle (Kundeninformationen)

3.4.3 Generierung der Produktdaten (Products Table)

Die Products-Table enthält Informationen zu Modeartikeln und simuliert eine Mischung aus erschwinglichen und luxuriösen Produkten.

Hierbei wurde sichergestellt, dass Preisnachlässe realistisch verteilt sind und bestimmte Marken als Luxusartikel klassifiziert wurden.

Datenfelder der Products-Table:

Attribute	Beschreibung
product_id	Eindeutige ID des Produkts
product_name	Kombination aus Farbe und Produkttyp
category	Kategorie des Produkts (Tops, Bottoms etc.)
brand	Marke des Produkts
material	Material des Produkts
price	Ursprungspreis des Produkts
stock	Verfügbare Menge

discount	Rabatt in % (40 % der Artikel haben einen Rabatt)
final_price	Endpreis nach Rabatt
luxury-brand	Gibt an, ob das Produkt eine Luxusmarke ist

Tabelle 3: Datenfelder der Produkte- Tabelle (Produktinformationen)

3.4.4 Generierung der Bestelldaten (Orders Table)

Die Orders-Table verbindet Kunden mit Produkten und enthält Informationen über Bestellungen, Retouren und Conversion-Rates.

Datenfelder der Orders-Table

Attribute	Beschreibung
Order_id	Eindeutige ID des Produkts
Product_id	Verknüpfung zur Produkts Table
User_id	Verknüpfung zur Users Table
Quantity	Anzahl der bestellten Einheiten
Total_Price	Gesamtpreis der Bestellung
Order_Status	Completed, Pending, Cancelled, Returned
Returns	Rückgabeanzahl (bei Returned-Bestellungen)
Return_rate	Anteil der retournierten Artikel
Timestamp	Zeit der Bestellung

Tabelle 4: Datenfelder der Orders-Tabelle (Bestellinformationen)

3.4.5 Kritische Reflexion der Datenqualität und Limitationen

Trotz der sorgfältigen Erstellung des synthetischen Datensatzes müssen bei der Interpretation der Analyseergebnisse gewisse Einschränkungen berücksichtigt werden. Synthetische Daten bieten eine innovative Möglichkeit, datenschutzkonforme Analysen durchzuführen, sind jedoch nicht frei von Schwächen. Insbesondere ergeben sich Herausforderungen im Hinblick auf **Datenqualität, Generalisierbarkeit und statistische Aussagekraft**.

1. Mangelnde Generalisierung und Verzerrungen

Ein wesentliches Problem synthetischer Daten besteht in ihrer begrenzten Generalisierbarkeit. Während einzelne Variablen realitätsnah erscheinen können, werden die komplexen Zusammenhänge zwischen diesen Variablen nicht immer adäquat abgebildet [12, S. 4]. Dies kann dazu führen, dass Business-Intelligence-Analysen auf fehlerhaften Annahmen beruhen und somit unzuverlässige Ergebnisse liefern.

2. Einschränkungen bei der statistischen Repräsentativität

Die Qualität synthetischer Daten wird oft anhand statistischer Metriken bewertet, jedoch sind diese Metriken nicht immer in der Lage, die gesamte Struktur eines Datensatzes realitätsgerecht wiederzugeben. Selbst wenn synthetische Daten den Originaldaten stark ähneln, können seltene Muster oder spezifische Merkmale möglicherweise nicht präzise nachgebildet werden, was systematische Verzerrungen zur Folge haben kann. [12, S. 1]

3. Datenschutzrisiken und Reidentifizierbarkeit

Ein weiteres Risiko synthetischer Datensätze besteht in der potenziellen **Reidentifizierbarkeit**. Platzer & Reutterer [12, S. 6] weisen darauf hin, dass synthetische Daten nicht lediglich eine modifizierte Version der Originaldaten darstellen dürfen, sondern vielmehr eigenständige, realistische Datensätze erzeugt werden müssen. Ihr entwickelter **Holdout-Based Privacy Risk Score** zeigt auf, dass viele Methoden zur Generierung synthetischer Daten eine unzureichende Generalisierung aufweisen und somit trotz Anonymisierung weiterhin Datenschutzrisiken bestehen bleiben. [12, S. 5]

4. Begrenzte Abbildung komplexer Zusammenhänge

Während synthetische Daten einzelne Variablen in vielen Fällen präzise nachbilden können, sind sie oft nicht in der Lage, komplexe mehrdimensionale Zusammenhänge korrekt abzubilden. Besonders in datenintensiven Bereichen wie dem **E-Commerce oder der Finanzbranche** kann dies zu problematischen Fehlinterpretationen führen, die weitreichende Auswirkungen auf datengetriebene Entscheidungen haben können. [12, S. 5]

5. Herausforderung des Trade-offs zwischen Datenschutz und Datenqualität

Ein zentrales Dilemma synthetischer Daten ist der **Privacy-Utility-Trade-off**, also die Herausforderung, eine optimale Balance zwischen Datenschutz und Datenqualität zu finden. Untersuchungen von Platzer & Reutterer [12, S. 1] zeigen, dass eine erhöhte Datensicherheit häufig mit einer Reduktion der analytischen Aussagekraft einhergeht. Bestehende synthetische Datengeneratoren weisen entweder eine zu hohe Ähnlichkeit zu den Originaldaten auf, wodurch **Datenschutzrisiken** entstehen, oder sie sind derart stark verfälscht, dass ihre **Nutzbarkeit für analytische Zwecke** eingeschränkt ist. [12, S. 2]

4. Einsatz von AWS-Diensten im Business-Intelligence-Szenario

4.1 Einleitung, warum AWS für Business Intelligence

Die Integration von Cloud-Technologien in datengetriebene Entscheidungsprozesse ist ein essenzielles Merkmal moderner Business-Intelligence-(BI)-Systeme. Amazon Web Services (AWS) bietet eine Vielzahl hochspezialisierter, skalierbarer und performanter Dienste zur Datenspeicherung, Verarbeitung, Analyse und Visualisierung. Laut einem Whitepaper von Infor bieten Cloud-basierte BI-Lösungen eine schnellere Kapitalrendite, geringere Implementierungskosten und verbesserte Zusammenarbeit, wodurch Unternehmen effizienter auf sich ändernde Marktbedingungen reagieren können. [13]

Im Vergleich zu anderen Cloud-Anbietern wie Google Cloud oder Microsoft Azure bietet AWS eine besonders tiefe Integration der Dienste. Beispielsweise lassen sich ETL-Prozesse mit AWS Glue nahtlos in Amazon Athena integrieren, was die Verarbeitungskosten reduziert. [11] (Amazon Athena)

Zudem ermöglicht AWS durch serverlose Technologien eine effiziente Nutzung von Ressourcen ohne die Notwendigkeit umfangreicher Infrastrukturverwaltung.

4.2 Amazon S3 – Speicherung und Verwaltung von Daten

4.2.1 Grundlagen von Amazon S3

Amazon S3 ist ein Cloud-basierter Objektspeicherdiensst, der es ermöglicht, Daten in beliebiger Menge sicher und skalierbar zu speichern. Unternehmen nutzen Amazon S3 für vielfältige Anwendungen wie Data Lakes, Websites, mobile Apps, Backups, Archivierungen, IoT-Geräte und Big-Data-Analysen. Der Dienst bietet Funktionen zur Datenorganisation, Zugriffskontrolle und Einhaltung von Compliance-Anforderungen. Mit seiner hohen Verfügbarkeit und Leistung unterstützt S3 Unternehmen dabei, ihre Daten effizient zu verwalten und zu schützen [11].

4.2.2 Anwendungsfall in dieser Arbeit

- **Rohdaten:** Ursprüngliche, unstrukturierte und halbstrukturierte E-Commerce-Daten im CSV-Format.

- **Transformierte Daten:** strukturierte, aggregierte und bereinigte Daten im optimierten Parquet-Format.
- **Analyseergebnisse:** exportierte KPI-Datensätze zur weiterführenden Visualisierung in QuickSight.

Die S3-Bucket-Struktur folgt einem durchdachten hierarchischen Datenmodell:

```
s3://data-fashion-ecommerce/
  └── raw-data/      # Ursprungliche CSV-Daten
  └── processed-data/ # Strukturierte Parquet-Daten
```

Die differenzierte Segmentierung der Datensätze innerhalb der S3-Bucket-Hierarchie ermöglicht eine effiziente Verwaltung, eine ressourcenschonende Verarbeitung und eine optimierte Abfrageleistung durch nachgelagerte Analyse-Dienste.

4.2.3 Vorteile von Amazon S3 in der BI-Architektur

Vorteile:

- Massive Skalierbarkeit für Big-Data-Analysen.
- Kosteneffiziente Speicheroptionen durch abgestufte Preisstrukturen (S3 Standard, S3 Infrequent Access, S3 Glacier).
- Nahtlose Integration mit AWS Glue, Athena und QuickSight zur End-to-End-Datenverarbeitung.

Limitationen:

- Keine direkte Query-Möglichkeit ohne externe Dienste wie Athena oder Redshift Spektrum.
- Netzwerklatenzen können Abfragen verlangsamen.

4.3 AWS Glue – Automatisierung von ETL-Prozessen

4.3.1 Grundlagen von AWS Glue

AWS Glue ist ein serverloser Datenintegrationsdienst von AWS, der es ermöglicht, Daten aus verschiedenen Quellen zu entdecken, aufzubereiten, zu transformieren und zu integrieren. Er unterstützt Anwendungsfälle wie Analysen, Maschine Learning und Anwendungsentwicklung. Mit AWS Glue können ETL-Pipelines (Extract, Transform, Load) visuell erstellt und überwacht sowie Daten in Data

Lakes oder Data Warehouses geladen werden. Der Dienst bietet einen zentralen Datenkatalog, der mit anderen AWS-Services wie Amazon S3, Athena und Redshift integriert werden kann. Dank der serverlosen Architektur entfällt die Notwendigkeit, Infrastruktur zu verwalten, wodurch AWS Glue flexibel und einfach zu nutzen ist. [11] (what is AWS Glue).

4.3.2 Anwendung in dieser Arbeit

AWS Glue wurde gezielt für folgende ETL-Prozesse eingesetzt:

- **Schema-Erkennung:** Glue-Crawler analysieren und erstellen automatisch Tabellen.
- **Datenbereinigung:** Entfernen von Duplikaten, Normalisierung der Datenstruktur.
- **Datenintegration:** Speicherung bereinigter Daten in Amazon S3 im Parquet-Format.

Durch den systematischen Einsatz von AWS Glue konnten die ETL-Prozesse vollständig automatisiert und die Betriebskosten erheblich reduziert werden.

4.3.3 Vorteile von AWS Glue für Business Intelligence

- **Automatisierte Schemaerkennung:** Glue-Crawler analysieren Datenquellen und generieren automatisch Tabellen.
- **Serverlose Architektur:** Keine Verwaltung von Servern oder Clustern erforderlich.
- **Flexibilität:** Unterstützung von PySpark für erweiterte Datenverarbeitung.

4.3.4 Kostenübersicht

AWS Glue wird pro ausgeführter ETL-Job-Laufzeit abgerechnet. Die Kosten basieren auf:

- **verarbeiteter Datenmenge (DPU-Stunden).**
- **Speicherverbrauch für den Glue Data Catalog.**

4.4 Amazon Athena – SQL-basierte Analyse von S3-Daten

4.4.1 Grundlagen von Amazon Athena

Amazon Athena ist ein serverloser, interaktiver Abfrageservice zur Analyse von Daten in Amazon S3. Er ermöglicht die Ausführung von **SQL-Abfragen** und **Apache Spark-Anwendungen** ohne Infrastrukturverwaltung. Nutzer können über die AWS Management Konsole oder APIs direkt auf S3-Daten zugreifen, Ad-hoc-SQL-Abfragen in Sekunden durchführen oder Spark-Code (Python/Scala) in integrierten Notebooks entwickeln. Athena skaliert automatisch, verarbeitet Abfragen parallel und liefert schnell Ergebnisse – selbst bei großen Datensätzen oder komplexen Operationen. Die Bezahlung erfolgt nutzungsbasiert (pro Abfrage/Spark-Session), wodurch Kosten nur bei aktiver Nutzung entstehen. [11] (was ist Amazon Athena)

4.4.2 Anwendung in dieser Arbeit

Amazon Athena wurde genutzt, um:

- SQL-Abfragen auf dem BI-Datenmodell (Sternschema) durchzuführen.
- Key Performance Indikatoren (KPIs) wie Umsatz, Retourenquote und Konversionsrate zu berechnen.
- Analyseergebnisse für die Visualisierung in Amazon QuickSight bereitzustellen.

4.4.3 Vorteile und Limitationen von Amazon Athena für Business Intelligence

- **Keine Infrastrukturverwaltung:** SQL-Abfragen können direkt auf Amazon-S3-Daten ausgeführt werden.
- **Einfache Integration:** Ergebnisse lassen sich direkt in Amazon QuickSight visualisieren.
- **Kosteneffizienz:** Abrechnung erfolgt nur für die verarbeiteten Datenmengen.
- **Limitationen:** Höhere Latenzzeiten im Vergleich zu spezialisierten Data Warehouses.

4.4.4 Vergleich mit klassischen Datawarehouses (Redshift)

Eigenschaft	Amazon Athena	Amazon Redshift
Infrastruktur	Serverlos	Clusterbasiert
Kostenmodell	Pay-per-Query	Pay-per-Instance
Leistung	Optimiert für Ad-hoc-Abfragen	Hohe Performance für komplexe Analysen
Skalierbarkeit	Automatisch	Manuelle Skalierung

Tabelle 5: Vergleich zwischen Amazon Athena und Amazon Redshift

4.5 Amazon QuickSight – Interaktive Datenvisualisierung

4.5.1 Grundlagen von Amazon QuickSight

Amazon QuickSight ist ein vollständig verwalteter, cloudbasierter BI-Dienst, der Daten aus verschiedenen Quellen zusammenführt und als interaktives Dashboard aufbereitet. Er bietet Enterprise-Sicherheit, globale Verfügbarkeit und integrierte Redundanz und ermöglicht so auch bei sehr großen Benutzerzahlen eine einfache, skalierbare Datenanalyse – ohne eigene IT-Infrastruktur verwalten zu müssen. [11] (was ist Amazon Quicksight?)

4.5.2 Anwendung in dieser Arbeit

Amazon QuickSight wurde verwendet, um:

- KPIs wie Umsatz, Retourenquote und Konversionsrate visuell darzustellen.
- Dynamische Dashboards zu erstellen, die direkt auf Amazon-Athena-Abfragen basieren.
- Analysen benutzerfreundlich und ohne tiefgehende SQL-Kenntnisse zu ermöglichen.

4.5.3 Vergleich mit anderen BI-Tools

Feature	Amazon QuickSight	Tableau	Power BI
Cloud-Native	Ja	Nein	Teilweise
Integriert mit AWS	Ja	Nein	Nein
Maschine Learning	Ja	Eingeschränkt	Eingeschränkt

Tabelle 6: Vergleich von Amazon QuickSight mit anderen BI-Tools

4.6 Fazit: Vorteile der AWS-Dienste für dieses BI-Szenario

Die Kombination aus Amazon S3, AWS Glue, Amazon Athena und Amazon QuickSight stellt eine leistungsfähige, skalierbare und kosteneffiziente Lösung für Business Intelligence dar:

- **Amazon S3** dient als zentrale Speicherplattform für Roh- und verarbeitete Daten.
- **AWS Glue** automatisiert die ETL-Prozesse und bereitet die Daten für Analysen vor.
- **Amazon Athena** ermöglicht SQL-basierte Analysen direkt auf S3-Daten.
- **Amazon QuickSight** visualisiert KPIs und Analyseergebnisse in interaktiven Dashboards.

Durch den Einsatz dieser AWS-Dienste konnte ein skalierbares und effizientes BI-System für den Fashion-E-Commerce entwickelt werden, das datengetriebene Entscheidungsprozesse optimiert.

5 Datenmodellierung und Implementierung

5.1 Erstellung des Datenmodells (Fakten- und Dimensionstabellen)

Eine effiziente Datenmodellierung bildet das Fundament für ein leistungsfähiges Business-Intelligence-(BI)-System. Ziel dieses Kapitels ist die detaillierte Darstellung der Datenstruktur, die im Rahmen dieser Arbeit für das Fashion-E-Commerce-Szenario entwickelt wurde. Dabei wird das zugrundeliegende **Sternschema** als bevorzugte Modellierungsstrategie erläutert.

Das Datenmodell dient dazu, große Mengen an E-Commerce-Daten effizient zu organisieren, um analytische Abfragen zu beschleunigen und fundierte Geschäftsentscheidungen zu ermöglichen. Besonders im BI-Bereich ist die Trennung zwischen **Dimensionstabellen** (beschreibende Informationen) und einer **Faktentabelle** (transaktionale Daten) von hoher Bedeutung.

5.2 Quellsystem-Datenmodell

Das Quellsystem basiert auf einer relationalen Struktur, in der folgende Kernentitäten enthalten sind:

- **Kunden** (Customers)
- **Produkte** (Products)
- **Bestellungen** (Orders)

Das folgende ER-Diagramm veranschaulicht die Struktur des Quellsystems:

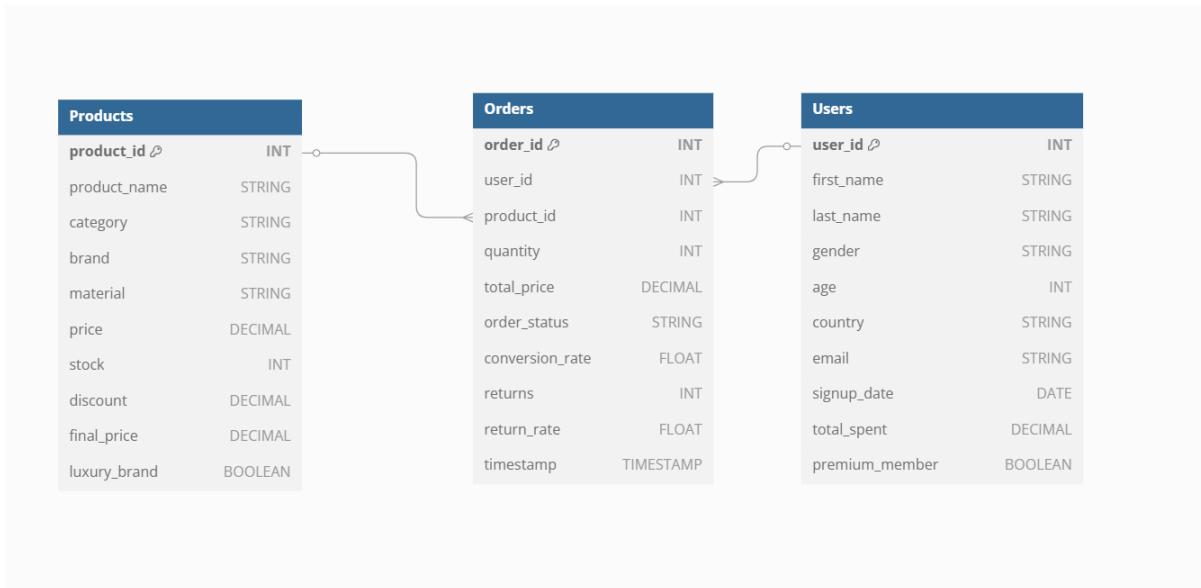


Abbildung 7: Entity-Relationship-Diagramm (ERD) des Quellsystems

Eigenschaften des Quellsystems

- **Normalisierte Struktur:** Die Daten sind in mehreren Tabellen aufgeteilt, um Redundanzen zu vermeiden.
- **Primär- und Fremdschlüssel:** Beziehungen zwischen Kunden, Bestellungen und Produkten sind über eindeutige IDs definiert.
- **Einfache, aber erweiterbare Struktur:** Zusätzliche Attribute wie Kundeninteraktionsdaten oder Lagerbestände könnten in Zukunft integriert werden.

5.3 Zielsystem-Datenmodell (Sternschema)

Zur effizienten analytischen Verarbeitung wurde das Quellsystem in ein **Sternschema** transformiert. Dieses Schema erlaubt schnelle Aggregationen und analytische Abfragen in Amazon Athena.

1- Faktentabelle: fact_orders

Die Faktentabelle bildet das **Herzstück des Sternschemas** und speichert die Kerntransaktionen der Plattform, insbesondere **Bestellungen, Umsätze und Retoureninformationen**.

Attribut	Typ	Beschreibung
order_id	bigint	Eindeutige ID der Bestellung (Primärschlüssel).
Userkey	bigint	Fremdschlüssel zur dim_users Tabelle.
date_id	bigint	Fremdschlüssel zur dim_time Tabelle (Bestelldatum).
Country	string	Land des Kunden (redundant für schnellere Abfragen).
Productkey	bigint	Fremdschlüssel zur dim_products Tabelle.
Category	string	Produktkategorie (Tops, Bottoms, Accessories etc.).
Year	int	Jahr der Bestellung.
Month	int	Monat der Bestellung.
Quantity	bigint	Anzahl der bestellten Produkte.
total_price	double	Gesamtpreis der Bestellung.
order_status	string	Status der Bestellung (<i>Completed, Pending, Cancelled, Returned</i>).

Tabelle 7: Struktur der Faktentabelle „fact_orders“ im Sternschema

Besonderheit:

- Die **Faktentabelle speichert die aggregierten Werte auf Bestellebene**, sodass Analysen über Umsatz, Bestellhäufigkeit und Retouren möglich sind.
- Durch die explizite Speicherung von *year* und *month* kann eine schnellere **zeitbasierte Analyse** durchgeführt werden.

2- Dimensionstabellen

Die Dimensionstabellen speichern **statische oder beschreibende Daten**, die mit der Faktentabelle verknüpft sind.

2.1 Kunden-Tabelle: dim_users

Diese Tabelle enthält Informationen über die **Kunden**, die Bestellungen auf der Plattform getätigt haben.

Attribut	Typ	Beschreibung
userkey	bigint	Primärschlüssel (eindeutige ID des Kunden).
first_name	string	Vorname des Kunden.
last_name	string	Nachname des Kunden.
country	string	Wohnsitzland des Kunden.
premium_member	boolean	True: Premium-Mitglied, False: Standard-Kunde.

Tabelle 8: Struktur der Dimensionstabelle „dim_users“ (Kundeninformationen)

Besonderheit:

- Premium-Kundenverhalten kann analysiert werden, um gezielte Marketingstrategien zu entwickeln.
- Länderbasierte Analysen ermöglichen Marktanteilsberechnungen.

2.2 Produkt-Tabelle: dim_products

Speichert **detaillierte Produktinformationen** inklusive Kategorie, Preis und Rabattstruktur.

Attribut	Typ	Beschreibung
productkey	bigint	Primärschlüssel (eindeutige ID des Produkts).
product_name	string	Name des Produkts.
category	string	Produktkategorie (z. B. Kleidung, Schuhe, Accessoires).
brand	string	Marke des Produkts.

Attribut	Typ	Beschreibung
price	double	Standardpreis des Produkts.
discount	double	Rabatt auf das Produkt.
final_price	double	Endpreis nach Rabattberechnung.
luxury_brand	boolean	True: Luxusmarke, False: Standardmarke.

Tabelle 9: Struktur der Dimensionstabelle „dim_products“ (Produktinformationen)

Besonderheit:

- Unterschiedliche Preis- und Rabattstrategien können analysiert werden.
- Luxusmarken vs. Standardmarken ermöglichen eine **Zielgruppenanalyse**.

2.3 Zeit-Tabelle: dim_time

Dient zur **zeitlichen Analyse von Bestellungen** und Umsätzen.

Attribut	Typ	Beschreibung
date_id	bigint	Primärschlüssel (eindeutige ID des Datums).
year	int	Jahr der Bestellung.
month	int	Monat der Bestellung.
day	int	Tag der Bestellung.

Tabelle 10: Struktur der Dimensionstabelle „dim_time“ (Zeitliche Analyse der Bestellungen und Umsätze)

Besonderheit:

- Ermöglicht **zeitliche Trends und Saisonanalysen**.
- Unterstützt Analysen zur **monatlichen Bestellwachstumsrate**.

5.4 Aufbau des Sternschemas für das BI-Szenario

Das **Datenmodell wurde als Sternschema** umgesetzt, um eine **effiziente Abfrageperformance** in analytischen Szenarien zu gewährleisten.

Die **Schlüsselaspekte des Sternschemas** sind:

- 1- **Klare Trennung von Fakten- und Dimensionstabellen**, wodurch Abfragen schneller ausgeführt werden können.
- 2- **Einfache Erweiterbarkeit**, falls neue Analysekriterien hinzukommen.
- 3- **Optimierung für OLAP-Abfragen**, da analytische Aggregationen direkt über die Faktentabelle durchgeführt werden können.

Beziehungen zwischen den Tabellen:

- **fact_orders.userkey → dim_users.userkey**
→ Ermöglicht **kundenbezogene Analysen**.
- **fact_orders.productkey → dim_products.productkey**
→ Ermöglicht **produktbezogene Analysen**.
- **fact_orders.date_id → dim_time.date_id**
→ Unterstützt **Saisonalitätsanalysen**.

Diese Verknüpfungen sorgen dafür, dass **alle Daten kontextuell miteinander in Beziehung gesetzt werden** können, wodurch die Effizienz und Flexibilität von BI-Abfragen maximiert wird.

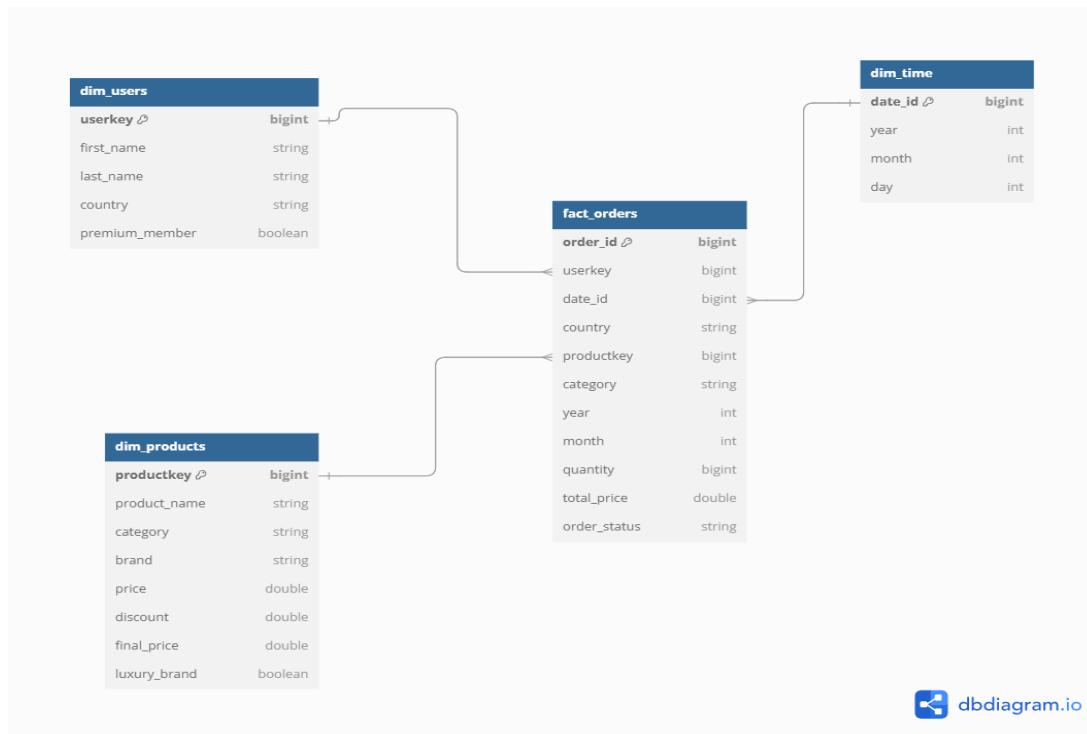


Abbildung 8: Entwurf des Sternschemas mit dbdiagram.io

5.5 Implementierung und Optimierung des Datenmodells

Speicherung in Amazon S3 und Athena

Die Daten werden in Amazon S3 gespeichert und über **Amazon Athena** abgefragt. Um die Leistung zu verbessern, wurde das **Parquet-Format** verwendet:

- **Warum Parquet?**
 - Spaltenbasiertes Speicherformat für schnellere Abfragen
 - Geringerer Speicherplatzbedarf im Vergleich zu CSV
 - Unterstützung von Komprimierungstechniken
- **Partitionierung & Indexierung**
 - Daten sind nach Jahr, Monat und Produktkategorie partitioniert.
 - Dies reduziert die zu scannende Datenmenge bei Abfragen erheblich.

6 Technische Umsetzung mit AWS-Diensten

Laut Buenabad-Chavez et al. (2024) bieten AWS-Instanzen eine effiziente Möglichkeit, komplexe Datenverarbeitungs-Workflows zu betreiben, ohne dass Benutzer Software lokal installieren oder große Datenmengen speichern müssen [14].

Die Umsetzung der **Business-Intelligence-(BI)-Lösung** erfolgt auf Basis einer **hochgradig skalierbaren, serverlosen Architektur**, die innerhalb der **Amazon-Web-Services-(AWS)-Cloud** bereitgestellt wird. Diese Architektur nutzt mehrere spezialisierte **Cloud-Dienste**, die darauf ausgelegt sind, **komplexe Datenanalysen mit minimalem Infrastrukturaufwand** zu ermöglichen.

Die Kernkomponenten der Architektur umfassen:

- **Amazon S3** – Zentrale Datenspeicherung für Roh- und transformierte Daten in einem skalierbaren Objektspeicher, der eine effiziente Verwaltung großer Datenmengen ermöglicht.
- **AWS Glue** – ein serverloses ETL-Framework, das die automatisierte Extraktion, Transformation und das Laden (ETL) von Daten übernimmt.
- **Amazon Athena** – ein SQL-basierter, interaktiver Abfragedienst, der es ermöglicht, große, in Amazon S3 gespeicherte Datensätze effizient auszuwerten.
- **Amazon QuickSight** – ein BI-Visualisierungstool, mit dem dynamische Dashboards erstellt werden können, um datenbasierte Erkenntnisse übersichtlich darzustellen.

- **IAM (Identity and Access Management)** – ein Sicherheitsframework zur Zugriffskontrolle und Verwaltung von Berechtigungen, das sicherstellt, dass nur autorisierte Benutzer auf Daten und Ressourcen zugreifen können.

Vergleich von AWS-Technologien mit Alternativen

Um die Wahl der eingesetzten Technologien zu begründen, wurden verschiedene **Cloud-Services** miteinander verglichen. Die folgende Tabelle zeigt einen **technischen Vergleich** zwischen den **eingesetzten AWS-Technologien** und **alternativen Lösungen** wie **Google BigQuery** und **Snowflake**:

Technologie	Vorteil	Nachteil
Amazon Athena	Serverlos, SQL-basiert, direkte Abfragen auf S3	Kosten steigen mit Abfragegröße, langsamere Performance als Redshift
Amazon Redshift	Hochperformantes Data Warehouse	Höhere Kosten, erfordert Wartung
Google BigQuery	Sehr schnelle Abfragen, stark für ML	Preisstruktur kann teuer werden
Snowflake	Automatische Skalierung, günstige Speicheroptionen	Komplexe Kostenmodelle

Tabelle 11: Vergleich von AWS-Technologien mit Alternativen

Dieser Vergleich soll aufzeigen, welche Vorteile und möglichen Einschränkungen die gewählten AWS-Dienste im Vergleich zu konkurrierenden Plattformen haben und inwieweit sie für die **spezifischen Anforderungen einer skalierbaren BI-Architektur** geeignet sind.

Entscheidung für AWS Glue und Athena: Diese wurden aufgrund der geringen Einstiegskosten, der einfachen Verwaltung und der Serverlosigkeit gewählt.

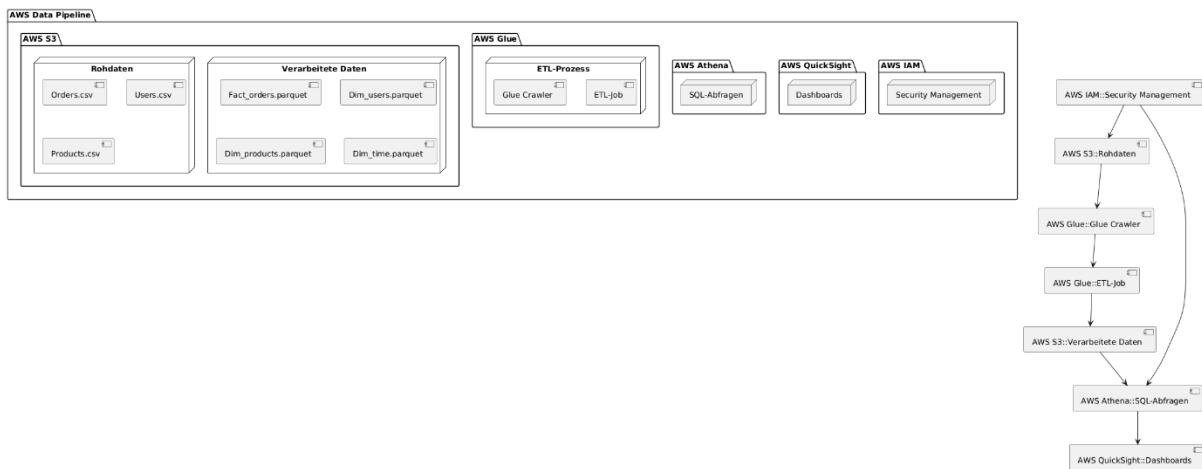


Tabelle 12: Architekturdiagramm der AWS-Datenpipeline

6.1 Datenspeicherung in Amazon S3

Amazon S3 (Simple Storage Service) ist ein skalierbarer Objektspeicherdienst von Amazon Web Services (AWS), der speziell für die sichere und effiziente Speicherung großer Datenmengen entwickelt wurde. Neben seiner hohen Skalierbarkeit und Verfügbarkeit zeichnet sich S3 durch eine nahtlose Integration mit anderen AWS-Diensten aus, was es zu einer zentralen Komponente moderner Cloud-basierter Datenverarbeitungslösungen macht. [11]

Hauptmerkmale von Amazon S3:

1. **Skalierbarkeit:** bietet eine nahezu unbegrenzte Skalierbarkeit und ermöglicht es Unternehmen, große Mengen an strukturierten und unstrukturierten Daten kosteneffizient zu speichern und zu verwalten [15]
2. **Zuverlässigkeit:** Die Datensicherheit in Amazon S3 wird durch eine Redundanz über mehrere physischen Standorte hinweg sichergestellt. Mit einer Datenhaltbarkeit von 99,999999999 % (11 Neunen) bietet der Dienst eine hohe Ausfallsicherheit und Schutz vor Datenverlust. [11]
3. **Kostenoptimierung:** Amazon S3 optimiert die Kosten durch verschiedene Speicherklassen, darunter S3 Glacier und S3 Intelligent-Tiering, die auf die jeweiligen Zugriffshäufigkeiten abgestimmt sind. Zudem ermöglicht das Pay-as-you-go-Modell in Kombination mit automatischen Speicherregeln (Lifecycle Policies) eine effiziente Kostenkontrolle. [11]

6.1.1 Erstellung des S3-Buckets

Eigenschaft des S3-Buckets:

Im Rahmen dieses Projekts wurde ein S3-Bucket mit folgenden spezifischen Eigenschaften erstellt:

- **Name:** data-fashion-E-Commerce

- **Region:** Frankfurt (eu-central-1)
- **Zugriff:** Zur Sicherstellung eines kontrollierten Zugriffs wurde eine **IAM-Rolle** eingerichtet, die ausschließlich AWS Glue und Amazon Redshift den Zugriff auf den Bucket erlaubt.

Rationale für die Nutzung von IAM-Rollen

Identity and Access Management (IAM) ist ein zentraler AWS-Dienst zur Verwaltung von Berechtigungen und Zugriffskontrollen. Durch den Einsatz von IAM-Rollen können Berechtigungen fein granular definiert werden, sodass ausschließlich autorisierte Dienste und Nutzer auf die Ressourcen zugreifen können. Im Gegensatz zu festen Benutzerkonten bieten IAM-Rollen eine flexiblere und sicherere Zugriffskontrolle, da sie zeitlich begrenzt zugewiesen und an spezifische AWS-Dienste gebunden werden können. [11]

Sicherheitsmaßnahmen: Zur Sicherstellung der Datensicherheit in diesem Projekt wurden die folgenden Maßnahmen implementiert, die auf den Best Practices von AWS basieren:

Serverseitige_Verschlüsselung: Alle im S3-Bucket gespeicherten Daten wurden mit dem AWS Key Management Service (KMS) verschlüsselt. Diese Maßnahme schützt Daten bei der Speicherung und erfüllt höchste Sicherheitsstandards, indem symmetrische Schlüssel sicher verwaltet werden. (AWS Security Practices) [11]

Bucket-Policies: Es wurden spezifische Zugriffsrichtlinien definiert, um den Zugriff ausschließlich autorisierten AWS-Diensten und Benutzern zu gestatten. Diese Policies minimieren das Risiko unbefugter Zugriffe und stellen sicher, dass Sicherheitsstandards eingehalten werden (AWS-Identity and Access Management). [11]

Zugriffsprotokollierung: Mit Amazon S3 Server Access Logging werden alle Zugriffsanfragen auf den S3-Bucket protokolliert. Diese Logs dienen der Nachverfolgbarkeit und Transparenz im Falle von Sicherheitsvorfällen (AWS Logging and Monitoring). [11]

Versionierung: Der S3-Bucket unterstützt die Versionierung, wodurch frühere Versionen von Objekten gespeichert und wiederhergestellt werden können. Diese Funktion schützt vor versehentlichem Löschen oder Überschreiben wichtiger Daten (AWS Versioning). [11]

6.1.2 Organisation der Rohdaten

Die Daten im S3-Bucket wurden in einer hierarchischen Struktur organisiert, um eine klare Trennung zwischen Roh- und verarbeiteten Daten zu gewährleisten. Diese Struktur dient dazu, die Verwaltung und Verarbeitung der Daten effizient zu gestalten sowie Fehlerquellen zu minimieren.

Struktur des Buckets:

raw-data/: Enthält die unveränderten Rohdaten, die direkt aus den Quellsystemen stammen.

The screenshot shows the AWS S3 console interface for the 'raw-data/' bucket. At the top, there are buttons for 'S3-URI kopieren' (Copy S3 URI) and 'Hochladen' (Upload). Below these are tabs for 'Objekte' (Objects) and 'Eigenschaften' (Properties), with 'Objekte' being selected. A search bar 'Objekte nach Präfix suchen' (Search objects by prefix) is present. A message states: 'Objekte sind die grundlegenden Entitäten, die in Amazon S3 gespeichert sind. Sie können Amazon S3 Inventory verwenden, um eine Liste aller Objekte in Ihrem Bucket abzurufen. Damit andere auf Ihre Objekte zugreifen können, müssen Sie ihnen explizit Berechtigungen erteilen. Weitere Informationen'.

Name	Typ	Letzte Änderung	Größe	Speicherklasse
orders/	Ordner	-	-	-
products/	Ordner	-	-	-
users/	Ordner	-	-	-

Abbildung 9: Verzeichnisstruktur des S3-Buckets für das Fashion-E-Commerce-Data-Warehouse vor dem ETL-Prozess

processed-data/: Enthält die transformierten und bereinigten Daten, die für weiterführende analytische Prozesse vorbereitet wurden.

The screenshot shows the AWS S3 console interface for the 'processed-data/' bucket. At the top, there are buttons for 'S3-URI kopieren' (Copy S3 URI) and 'Hochladen' (Upload). Below these are tabs for 'Objekte' (Objects) and 'Eigenschaften' (Properties), with 'Objekte' being selected. A search bar 'Objekte nach Präfix suchen' (Search objects by prefix) is present. A message states: 'Objekte sind die grundlegenden Entitäten, die in Amazon S3 gespeichert sind. Sie können Amazon S3 Inventory verwenden, um eine Liste aller Objekte in Ihrem Bucket abzurufen. Damit andere auf Ihre Objekte zugreifen können, müssen Sie ihnen explizit Berechtigungen erteilen. Weitere Informationen'.

Name	Typ	Letzte Änderung	Größe	Speicherklasse
dim_products/	Ordner	-	-	-
dim_time/	Ordner	-	-	-
dim_users/	Ordner	-	-	-
fact_orders/	Ordner	-	-	-
Unsaved/	Ordner	-	-	-

Abbildung 10: Verzeichnisstruktur des S3-Buckets für den ETL-Prozess

Optimierung der Datenverarbeitung durch die Nutzung von Parquet-Dateien

Die Verwendung von Parquet-Dateien, einem binären, spaltenorientierten Dateiformat, ermöglicht eine effiziente Datenexploration in Data Lakes, indem aufwendige ETL-Prozesse vermieden werden. Fortschrittliche Techniken wie die inkrementelle Sammlung von Abfragestatistiken und die Segmentierung von Dateien in handhabbare Abschnitte verbessern die Leistung und ermöglichen die selektive Datenabfrage. Dadurch wird die Effizienz der Abfrageausführung erheblich gesteigert. [16]

Eigenschaft	CSV	Parquet
Speicherformat	Zeilenbasiert	Spaltenbasiert
Speicherplatzbedarf	Hoch	Reduziert um 30-50%

Query-Geschwindigkeit (Athena)	Langsam	Optimiert durch Spaltenzugriff
--------------------------------	---------	--------------------------------

Tabelle 13: Vergleich zwischen CVS- und Parquet-Formaten

Performance-Analyse für Athena-Abfragen:

- **CSV-Datei (100 GB):** Abfragekosten = **\$0,50** (gescannte Datenmenge: 100 GB).
- **Parquet-Datei (30 GB):** Abfragekosten = **\$0,15** (gescannte Datenmenge: 30 GB).
- **Ergebnis:** Durch die Nutzung von Parquet können **bis zu 70 % Kosten gespart werden.**

6.1.3 Ergebnisse

Die Dateien orders.csv, users.csv und products.csv wurden erfolgreich in den Ordner raw-data/ hochgeladen. Diese Trennung stellt sicher, dass die Rohdaten unverändert und leicht zugänglich für weitere Verarbeitungsschritte bleiben. Die vorbereiteten Daten sind nun für den ETL-Prozess mit AWS Glue verfügbar.

6.2 ETL-Prozesse mit AWS Glue

AWS Glue ist ein serverloser Datenintegrationsdienst, der es ermöglicht, Daten aus mehr als 70 verschiedenen Quellen zu erkennen, zu bereinigen, zu transformieren und zentral zu katalogisieren. Der Service bietet umfassende Unterstützung für ETL- und ELT-Workloads sowie Streaming-Daten und erleichtert die Integration in AWS-Dienste wie Amazon S3, Amazon Redshift und Amazon Athena. Mit dem Glue Data Catalog können Metadaten zentral verwaltet werden, was die Suche und Abfrage von Daten erheblich vereinfacht. Die serverlose Architektur von AWS Glue eliminiert die Notwendigkeit, Infrastruktur zu verwalten, und ermöglicht durch benutzerfreundliche Tools sowohl Entwicklern als auch Geschäftsanwendern die einfache Erstellung und Verwaltung von Datenpipelines. Dieser Ansatz reduziert den Aufwand für die Datenintegration und bietet eine flexible Lösung für Analysen, maschinelles Lernen und Anwendungsentwicklung (AWS Glue). [11]

Laut der Studie von Saxena et al. (2023) ist AWS Glue ein serverloser Dienst, der speziell entwickelt wurde, um die Extraktion, Bereinigung, Anreicherung, Transformation und Organisation von Daten effizient und kosteneffektiv zu gestalten. Glue verwendet serverlose Apache Spark- und Python-Engines, die durch einen eigens entwickelten Ressource Manager unterstützt werden, um einen schnellen Start und automatische Skalierung zu gewährleisten. [17, S. 3357]

6.2.1 Einrichtung des Glue-Crawlers

6.2.1.1 Einrichtung des Crawlers für raw_data

Einleitung:

Laut der Studie von Saxena et al. (2023) bieten Glue-Crawler eine effiziente Möglichkeit, die Struktur von Datenquellen automatisch zu erkennen und Metadaten im Glue Data Catalog zu speichern. Diese

Crawler wurden entwickelt, um die Herausforderungen bei der Arbeit mit semi-strukturierten und sich dynamisch ändernden Daten, wie z. B. JSON-Logs, zu lösen. Sie eliminieren die Notwendigkeit manueller Konfiguration und ermöglichen eine automatisierte Verwaltung von Metadaten, was sie besonders wertvoll für Data-Lake-Umgebungen macht. [17, S. 3567]

Zielsetzung:

Das Ziel des Crawlers ist es, die Rohdaten aus dem S3-Bucket zu erkennen und automatisch eine Datenbank im Glue Data Catalog mit strukturierten Tabellen zu erstellen. Dies dient als Basis für den ETL-Prozess und die anschließende Datenanalyse.

1. Datenquelle analysieren und vorbereiten:

Die Rohdaten werden im S3-Bucket gespeichert und dienen als Eingabe für den Glue-Crawler. Eine strukturierte Organisation der Daten in Ordnern erleichtert die Verarbeitung. Für dieses Projekt befinden sich die Rohdaten im Pfad:

```
s3://data-fashion-ecommerce/raw-data/
```

2. Definition der Datenstruktur

Die zu analysierenden Daten umfassen mehrere Attribute, darunter numerische, kategoriale und zeitliche Daten. Die Zielstruktur des Glue Crawlers besteht darin, diese Attribute als Tabellen mit spezifischen Datentypen zu klassifizieren und im Data Catalog zu speichern. Ziel ist eine flexible Strukturierung der Daten für spätere Transformationen.

3. Einrichtung des Glue Crawlers

Die Einrichtung eines Glue Crawlers erfolgt in mehreren Schritten:

3.1 Erstellung des Crawlers

- **Crawler-Name:** Ein eindeutiger Name wird vergeben, z. B. raw_data.
- **Zugriff auf Datenquellen:** Der Crawler wird so konfiguriert, dass er den S3-Bucket durchsucht, einschließlich aller Unterverzeichnisse.

Set crawler properties

Crawler details [Info](#)

Name
raw_data
Name can be up to 255 characters long. Some character set including control characters are prohibited.

Description - optional
Enter a description
Descriptions can be up to 2048 characters long.

Tags - optional
Use tags to organize and identify your resources.

Abbildung 11: Konfiguration des Crawlers in AWS Glue zur automatischen Erkennung von S3-Daten

3.2 Auswahl der Datenbank

- **Datenbankname:** Eine neue Datenbank wird erstellt, um die Tabellen aufzunehmen. Beispiel: ecommerce_raw_db.
- **Speicherung der Metadaten:** Der Data Catalog speichert alle erkannten Schemata und stellt sie für Abfragen bereit.

Choose data sources and classifiers

Data source configuration

Is your data already mapped to Glue tables?

Not yet
Select one or more data sources to be crawled.

Yes
Select existing tables from your Glue Data Catalog.

Data sources (1) [Info](#)

The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
S3	s3://data-fashion-e-commerce/raw-data/	Recrawl all

Custom classifiers - optional

A classifier checks whether a given file is in a format the crawler can handle. If it is, the classifier creates a schema in the form of a StructType object that matches that data format.

Custom classifiers [Info](#)

Select one or more classifiers to use with this crawler.

Choose one or more classifiers

Abbildung 12: Auswahl der Datenquelle und Klassifikatoren für den AWS-Glue-Crawler

3.3 IAM-Rollen und Berechtigungen

- **Rollenanforderungen:** Eine IAM-Rolle mit minimalen Berechtigungen wird definiert. Diese Rolle ermöglicht dem Crawler den Zugriff auf den S3-Bucket und die Glue-Dienste.
- **Berechtigungen:** Die Rolle enthält s3:GetObject, s3>ListBucket für den spezifischen Bucket und glue:/* für den Zugriff auf Glue-Ressourcen.

Configure security settings

IAM role [Info](#)
Existing IAM role
AWSGlueServiceRole-S3 [View](#) [Edit](#)
[Create new IAM role](#) [Update chosen IAM role](#)

Only IAM roles created by the AWS Glue console and have the prefix "AWSGlueServiceRole-" can be updated.

Lake Formation configuration - optional
Allow the crawler to use Lake Formation credentials for crawling the data source. [Learn more](#).
 Use Lake Formation credentials for crawling S3 data source
Checking this box will allow the crawler to use Lake Formation credentials for crawling the data source. If the data source is registered in another account, you must provide the registered account ID. Otherwise, the crawler will crawl only those data sources associated to the account. Only applicable to S3, Glue Catalog, Iceberg, and Hudi data sources.

► Security configuration - optional
Enable at-rest encryption with a security configuration.

Abbildung 13: Konfiguration der IAM-Rolle für den Zugriff auf den S3-Bucket und AWS Glue

3.4 Konfiguration des Crawlers

- Frequenz der Ausführung:** Der Crawler wird auf Abruf ausgeführt, um Kosten zu minimieren.
- Erkennung von Schemata:** Der Crawler analysiert die Daten und extrahiert automatisch deren Struktur, einschließlich Datentypen und Beziehungen.

Step 1: Set crawler properties [Edit](#)
Set crawler properties
Name raw_data Description - Tags -

Step 2: Choose data sources and classifiers [Edit](#)
Data sources (1) [Info](#)
The list of data sources to be scanned by the crawler.

Type	Data source	Parameters
S3	s3://data-fashion-commerce/raw-data/	Recrawl all

Classifiers (1) [Info](#)
A classifier can help determine the schema of your data.

Name	Type	Classification	Last updated (UTC)
csv_classifier	CSV	-	February 5, 2025 at 15:35:40

Step 3: Configure security settings [Edit](#)
Configure security settings
IAM role AWSGlueServiceRole-S3 Security configuration - Lake Formation configuration -

Step 4: Set output and scheduling [Edit](#)
Set output and scheduling
Database fashion_ecommerce_db Table prefix - optional Maximum table threshold - optional Schedule On demand

Abbildung 14: Übersicht über die Konfiguration des AWS-Glue-Crawlers für das Fashion-E-Commerce-Data-Warehouse

4. Ergebnisse des Crawlers

Nach der erfolgreichen Ausführung des Crawlers wird eine strukturierte Datenbank erstellt. Diese enthält Tabellen, die die Attribute der Rohdaten repräsentieren. Beispiele:

- **Tabellen:** orders.csv, products.csv, users.csv

Die erstellten Tabellen bieten eine standardisierte Basis für Transformationen im weiteren ETL-Prozess.

#	Column name	Data type
1	order_id	bigint
2	user_id	bigint
3	product_id	bigint
4	quantity	bigint
5	total_price	double
6	order_status	string
7	conversion_rate	double
8	returns	bigint
9	return_rate	double
10	timestamp	string

Abbildung 15: Extrahierte Tabellenstruktur der Bestellungen (orders.csv) nach der Crawler-Ausführung

#	Column name	Data type
1	product_id	bigint
2	product_name	string
3	category	string
4	brand	string
5	material	string
6	price	double
7	stock	double
8	discount	double
9	final_price	double
10	luxury_brand	boolean

Abbildung 16: Extrahierte Tabellenstruktur der Produkte (Products) nach der Crawler-Ausführung

#	Column name	Data type
1	user_id	bigint
2	first_name	string
3	last_name	string
4	gender	string
5	age	double
6	country	string
7	email	string
8	signup_date	string
9	total_spent	double
10	premium_member	boolean

Abbildung 17: Extrahierte Tabellenstruktur der Nutzer (Users) nach der Crawler-Ausführung

Problemstellung

Während der Implementierung des AWS Glue Crawlers wurde ursprünglich der gesamte „raw-data“-Ordner als Datenquelle angegeben:

s3://data-fashion-e-commerce/raw-data/

Die Daten (CSV-Dateien für Bestellungen, Produkte und Benutzer) wurden dabei direkt in diesem Ordner abgelegt, ohne separate Unterverzeichnisse für jede Datei. Nach der Ausführung des Crawlers wurden die Tabellen in Amazon Athena zwar mit den richtigen Spaltennamen, aber ohne Datensätze erstellt. Dies deutete darauf hin, dass der Crawler die CSV-Dateien nicht richtig verarbeitet hat.

Ursache des Problems

AWS Glue Crawlers sind so konzipiert, dass sie gesamte Verzeichnisse durchsuchen und die darin enthaltenen Datenquellen basierend auf der Dateistruktur erkennen. Wenn jedoch mehrere CSV-Dateien direkt in einem einzigen Ordner gespeichert sind, kann der Crawler Schwierigkeiten haben, die Daten korrekt zu identifizieren und in die entsprechenden Tabellen zu laden.

Das Hauptproblem bestand darin, dass alle CSV-Dateien in einem einzigen Verzeichnis gespeichert waren, anstatt in separaten Unterverzeichnissen. Dies führte dazu, dass AWS Glue den Inhalt der Dateien nicht richtig zuordnen konnte, weshalb die Tabellen zwar erstellt, aber ohne Datensätze befüllt wurden.

Lösung: Erstellung separater Unterverzeichnisse für jede CSV-Datei

Um das Problem zu beheben, wurde die Verzeichnisstruktur in **Amazon S3** überarbeitet, indem für jede CSV-Datei ein eigener Unterordner erstellt wurde:

S3-Bucket: data-fashion-E-Commerce

-  **raw-data/** (*Hauptordner, der an den Crawler übergeben wurde*)
 -  **orders/** → Enthält orders.csv
 -  **products/** → enthält products.csv
 -  **users/** → enthält users.csv

Anpassung und erneute Konfiguration des AWS-Glue-Crawlers

Nach der Umstrukturierung der Datenarchitektur wurde eine Anpassung des AWS-Glue-Crawlers vorgenommen, um die Datenverarbeitung zu optimieren. Zuvor waren alle CSV-Dateien direkt im übergeordneten Verzeichnis `s3://data-fashion-ecommerce/raw-data/` abgelegt. Diese Konfiguration führte dazu, dass AWS Glue die Daten nicht korrekt erkannte, was dazu führte, dass die Tabellen in Amazon Athena zwar erstellt wurden, aber keine Datensätze enthielten.

Um dieses Problem zu beheben, wurde die Verzeichnisstruktur umorganisiert, indem separate Unterverzeichnisse für jede Datendatei erstellt wurden. Die neue Struktur gliederte sich wie folgt:

- `s3://data-fashion-ecommerce/raw-data/orders/` (enthält orders.csv)
- `s3://data-fashion-ecommerce/raw-data/products/` (enthält products.csv)
- `s3://data-fashion-ecommerce/raw-data/users/` (enthält users.csv)

Im Zuge dieser Anpassung wurde der AWS-Glue-Crawler neu konfiguriert, wobei weiterhin das übergeordnete Verzeichnis (`s3://data-fashion-ecommerce/raw-data/`) als Datenquelle angegeben wurde, ohne explizit auf die einzelnen Unterverzeichnisse zu verweisen. AWS Glue erkannte daraufhin automatisch die Struktur der einzelnen Datensätze und erstellte für jede Kategorie eine eigene Tabelle.

Erneute Ausführung und erfolgreiche Erkennung der Daten

Nach der Anpassung wurde der Crawler erneut ausgeführt. Dabei wurde bestätigt, dass AWS Glue die Unterverzeichnisse ordnungsgemäß durchsuchen und die darin enthaltenen CSV-Dateien korrekt identifizieren konnte. Diese Änderung führte dazu, dass die Datenstruktur nun in den AWS-Glue-Data-Catalog integriert wurde und in Amazon Athena vollständige Tabellen mit den erwarteten Datensätzen verfügbar waren.

Validierung in Athena

Zur Überprüfung wurde folgende **SQL-Abfrage in Athena** ausgeführt:

```
SELECT * FROM orders LIMIT 10;
```

Ergebnis:

- Die Abfrage lieferte **korrekte Datensätze** zurück.
- **Vorherige Probleme mit leeren Tabellen wurden vollständig behoben.**

Fazit

Die Lösung bestand darin, für jede CSV-Datei ein eigenes Unterverzeichnis zu erstellen, anstatt alle Dateien direkt im **raw-data/-Verzeichnis** zu speichern. Dies ermöglichte dem AWS-Glue-Crawler, die Daten **korrekt zu erkennen und zu verarbeiten**.

Vorteile dieser Lösung:

- **Strukturierte Datenorganisation:** Jedes Dataset ist in einem eigenen Ordner abgelegt.
- **Automatische Erkennung neuer Daten:** AWS Glue erkennt neue Dateien automatisch in den entsprechenden Unterverzeichnissen.
- **Korrekte Datenaufnahme in Athena:** Tabellen werden nun mit den richtigen Spalten und Datensätzen befüllt.

Diese Erkenntnisse wurden erfolgreich genutzt, um die **ETL-Prozesse weiter zu optimieren** und sicherzustellen, dass AWS Glue und Amazon Athena reibungslos zusammenarbeiten.

6.2.1.2 Einrichtung des Crawlers für Processed Data

1. Zielsetzung und Datenquelle:

Der Crawler für die verarbeiteten Daten wurde erstellt, um alle in S3-Buckets gespeicherten Dimensionstabellen und die Faktentabelle zu katalogisieren.

- Pfad der Datenquelle: s3://data-fashion-eCommerce/processed-data/
- Zieldatenbank: fashion_ecommerce_db

2 Struktur der verarbeiteten Daten

Die verarbeiteten Daten umfassen mehrere Tabellen:

1. dim_products

- Attribute: productkey, product_name, category, brand, price, discount, final_price, luxura_brand

2. **dim_time**

- Attribute: date_id, year, month, day

3. **dim_users**

- Attribute: user_id, first_name, last_name, gender, age, country, email, signup_date, total_spent, premium_member

4. **Fact_orders**

- Attribute: order_id, productkey, userkey, country, category, date_id, year, month, quantity, totalprice, order_status

3. Einrichtung des Processed-data-Crawlers

Schritte zur Einrichtung:

1. Crawler erstellen:

- Name: processed_data
- Datenquelle: s3://data-fashion-eCommerce/processed-data/
- Datenbank: fashion_ecommerce_db

2. IAM-Rollen und Berechtigungen:

- Die gleiche IAM-Rolle (AWSGlueServiceRole-s3) wie beim RawDataCrawler wurde genutzt.

3. Konfiguration:

- Häufigkeit der Ausführung: Auf Abruf
- Schema-Erkennung: Automatisch

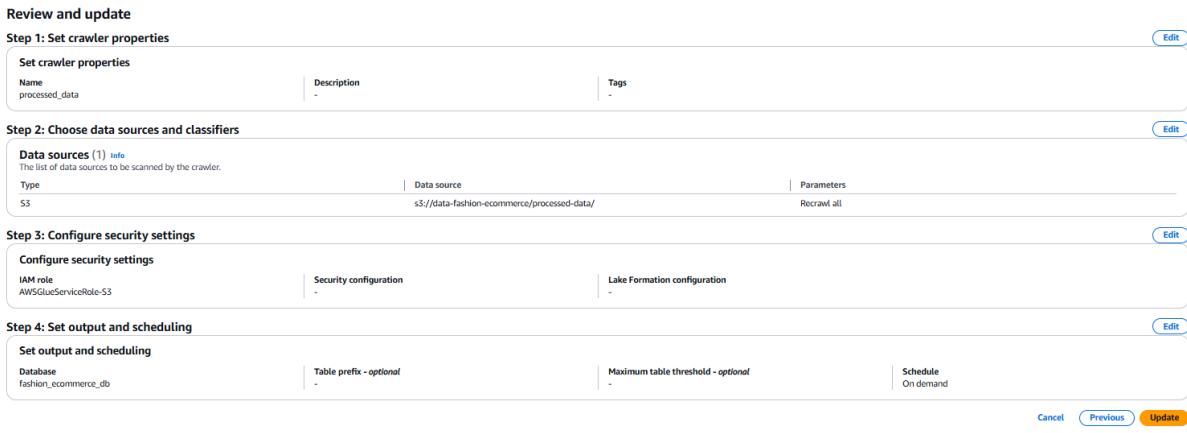


Abbildung 18: Übersicht über den Crawler processed_data_crawler

Ergebnisse:

Nach der erfolgreichen Ausführung des Crawlers wurden alle Dimensionstabellen und die Faktentabelle in der ecommerce_processed_db katalogisiert.

<input type="checkbox"/> Name	▲ Database	▼ Location	▼ Classification
<input type="checkbox"/> dim_products	fashion_ecommerce_db	s3://data-fashion-commerce/processed	Parquet
<input type="checkbox"/> dim_time	fashion_ecommerce_db	s3://data-fashion-commerce/processed	Parquet
<input type="checkbox"/> dim_users	fashion_ecommerce_db	s3://data-fashion-commerce/processed	Parquet
<input type="checkbox"/> fact_orders	fashion_ecommerce_db	s3://data-fashion-commerce/processed	Parquet

Abbildung 19: Übersicht über ecommerce_processed_db nach Ausführung des Crawlers

4. Problemstellung: Unerwartete Erstellung mehrerer Tabellen durch den AWS-Glue-Crawler

Während der initialen Verarbeitung der Daten im processed-data-Bereich in Amazon S3 führte die Ausführung des AWS-Glue-Crawlers zu einer unerwarteten Erstellung von über 66 Tabellen anstelle der erwarteten vier Haupttabellen (fact_orders, dim_users, dim_products, dim_time). Dieses Problem erschwerte die strukturierte Abfrage der Daten in Amazon Athena und führte zu Redundanzen im AWS Glue Data Catalog.

Analyse der Ursache

Die Ursache des Problems lag in der ursprünglichen Implementierung des ETL-Prozesses, bei der die Tabellen in partitionierter Form gespeichert wurden. Apache Spark, welches von AWS Glue genutzt wird, speichert partitionierte Daten in mehreren Unterverzeichnissen mit separaten Parquet-Dateien. Da der AWS-Glue-Crawler die Ordnerstruktur als separate Datenquellen interpretierte, wurden für jede Partition eigenständige Tabellen erstellt, was zur erheblichen Vervielfachung der Tabelleneinträge führte.

Lösung: Anpassung des ETL-Prozesses zur Vermeidung von Partitionierungen

Um dieses Problem zu beheben, wurde die Speicherstrategie im ETL-Prozess angepasst. Statt der partitionierten Speicherung wurde nun für jede Tabelle eine einzelne Datei abgelegt.

Die Anpassungen umfassten folgende Änderungen:

Modifikation der Speichermethode:

Die partitionierte Speicherung (`.write.mode("overwrite").parquet(...)`) wurde entfernt.

Stattdessen wurde eine einzelne Parquet-Datei für jede Tabelle gespeichert, indem die Daten explizit in einem separaten temporären Ordner zwischengespeichert und anschließend als einzelne Datei umbenannt wurden.

Beibehaltung der übergeordneten S3-Pfadstruktur:

Der AWS-Glue-Crawler wurde nicht auf einzelne Tabellen konfiguriert, sondern weiterhin auf das übergeordnete Verzeichnis `s3://data-fashion-ecommerce/processed-data/` angewendet.

AWS Glue konnte nun die Struktur korrekt interpretieren und erkannte jede Tabelle als einzelne Einheit, anstatt mehrere Partitionen zu erstellen.

Ergebnis und Validierung

- Nach der Anpassung des ETL-Prozesses wurde der AWS-Glue-Crawler erneut ausgeführt.
 - Es wurden nun genau vier Tabellen (`fact_orders`, `dim_users`, `dim_products`, `dim_time`) im AWS-Glue-Data-Catalog registriert.
 - Die Daten konnten erfolgreich in Amazon Athena abgefragt werden.
 - Redundante Tabellen und Partitionen wurden vollständig vermieden.

Validierung durch SQL-Abfrage in Athena:

```
SELECT * FROM fact_orders LIMIT 10;
```

→ Ergebnis: Die erwarteten Datensätze wurden erfolgreich zurückgegeben.

Fazit

Durch die gezielte Optimierung des ETL-Prozesses wurde das Problem der übermäßigen Tabellenbildung in AWS Glue erfolgreich behoben. Die Speicherung als eine einzelne Datei pro Tabelle optimierte die Datenverwaltung und ermöglichte eine effizientere Abfrage der Daten in Amazon Athena.

6.2.2 Durchführung des ETL-Jobs

Zielsetzung:

Der **ETL-Prozess (Extract, Transform, Load)** wurde implementiert, um Rohdaten aus dem Data Lake in **Amazon S3** in ein standardisiertes Sternschema zu überführen. Ziel ist die **Bereitstellung einer sauberen, strukturierten und performanten Datenbasis** für Business-Intelligence-Analysen in Amazon Athena und Visualisierungen in **Amazon QuickSight**.

Durch den ETL-Prozess werden folgende Ziele verfolgt:

- Extraktion der Rohdaten aus den Quelldatenbanken in AWS Glue.
- Transformation der Daten in ein relationales Modell mit Faktentabelle und Dimensionstabellen.
- Bereinigung und Normalisierung der Daten (z. B. Entfernen von Duplikaten, Anpassung von Formaten).
- Speicherung der bereinigten Daten in Amazon S3 im Parquet-Format zur Optimierung von Abfragen.
- Reduzierung der Speicherplatzanforderungen durch die Speicherung einer einzigen Datei pro Tabelle, um redundante Partitionierungen zu vermeiden.

ETL-Prozess – detaillierte Schritte für jede Tabelle

Der ETL-Prozess gliedert sich in drei Hauptphasen:

1. **Extraktion (Extract):** Laden der Daten aus den Rohdatenquellen in AWS Glue.
2. **Transformation (Transform):** Verarbeitung, Bereinigung und Modellierung der Daten in das Sternschema.
3. **Laden (Load):** Speicherung der bereinigten Daten als **eine einzelne Datei pro Tabelle in Amazon S3**.

Nachfolgend werden die einzelnen ETL-Schritte für die Faktentabelle (`fact_orders`) und die Dimensionstabellen (`dim_users`, `dim_products`, `dim_time`) erläutert.

6.2.2.1 Extraktion der Daten

Die Rohdaten für Bestellungen, Kunden und Produkte werden aus den **AWS-Glue-Datenkatalogtabellen** extrahiert. Hierbei werden die Daten direkt aus **Amazon S3** geladen.

Code-Snippet für die Extraktion:

```
# Initialize Spark and Glue Context
spark = SparkSession.builder.appName("FashionEcommerceETL").getOrCreate()
glueContext = GlueContext(spark.sparkContext)
```

```
# Extract raw data from AWS Glue Catalog
orders_df = glueContext.create_dynamic_frame. from_catalog(database="fashion_ecommerce_db",
table_name="orders").toDF()
users_df = glueContext.create_dynamic_frame. from_catalog(database="fashion_ecommerce_db",
table_name="users").toDF()
products_df = glueContext.create_dynamic_frame. from_catalog(database="fashion_ecommerce_db", table_name="products").toDF()
```

- Die Rohdaten werden nun als DataFrames bereitgestellt und können weiterverarbeitet werden.

6.2.2.2 Transformation der Daten in Dimensionstabellen

1. Erstellung der Zeitdimension (dim_time)

Die dim_time-Tabelle wird aus der Spalte timestamp der Bestellungen generiert, um Analysen auf Jahres-, Monats- und Tagesebene zu ermöglichen.

Transformation für dim_time:

```
from pyspark.sql.functions import year, month, dayofmonth, date_format
dim_time_df = orders_df.select(
    date_format(orders_df.timestamp, "yyyyMMdd").cast("string").alias("date_id"),
    year(orders_df.timestamp).alias("year"),
    month(orders_df.timestamp).alias("month"),
    dayofmonth(orders_df.timestamp).alias("day")
).distinct()
```

- dim_time ermöglicht zeitbasierte Analysen und KPI-Berechnungen wie Monatsumsätze und Wachstumsraten.

2. Erstellung der Produktdimension (dim_products)

Die dim_products-Tabelle enthält alle statischen Produktinformationen, wie Name, Marke, Preis und Rabatt.

Transformation für dim_products:

```
dim_products_df = products_df.select(
    products_df.product_id.alias("productkey"),
    products_df.product_name,
```

```

products_df.category,
products_df.brand,
products_df.price,
products_df.discount,
products_df.final_price,
products_df.luxury_brand
)

```

- dim_products ermöglicht detaillierte Analysen zu Produktkategorien, Preisstrategien und Rabatten.

3. Erstellung der Kundendimension (dim_users)

Die dim_users-Tabelle enthält Informationen über Kunden, darunter Premium-Status und Wohnsitzland.

Transformation für dim_users:

```

dim_users_df = users_df.select(
    users_df.user_id.alias("userkey"),
    users_df.first_name,
    users_df.last_name,
    users_df.country,
    users_df.premium_member
)

```

- dim_users wird für Kundenanalysen, Marktsegmentierung und Premium-Mitglieder-Tracking genutzt.

6.2.2.3 Transformation der Faktentabelle (fact_orders)

Die fact_orders-Tabelle aggregiert die **Bestelldaten** mit den Dimensionen **Zeit, Kunde und Produkt**.

Transformation für fact_orders:

```

fact_orders_df = orders_df \
    .withColumn("date_id", date_format(orders_df.timestamp, "yyyyMMdd").cast("string")) \

```

```

.join(dim_users_df, orders_df.user_id == dim_users_df.userkey, "left") \
.join(dim_products_df, orders_df.product_id == dim_products_df.productkey, "left") \
.join(dim_time_df, "date_id", "left") \
.select(
    orders_df.order_id.alias("order_id"),
    dim_users_df.userkey,
    dim_users_df.country,
    dim_products_df.productkey,
    dim_products_df.category,
    dim_time_df.date_id,
    dim_time_df.year,
    dim_time_df.month,
    orders_df.quantity,
    orders_df.total_price,
    orders_df.order_status
)

```

- fact_orders ist die zentrale Faktentabelle für **Umsatzanalysen, Retourenquoten und Kundensegmentierung.**

6.2.2.4 Laden der bereinigten Daten in Amazon S3

Die Daten werden **ohne Partitionierung** als einzelne **Parquet-Dateien** gespeichert, um die Effizienz des Glue Crawlers zu gewährleisten.

Laden der Daten in S3:

```

import boto3

common_s3_path = "s3://data-fashion-e-commerce/processed-data/"

def save_as_single_file(df, table_name):
    """ Speichert DataFrame als einzelne Datei in S3 """

    temp_path = f'{common_s3_path}{table_name}/temp_output'

    final_path = f'{common_s3_path}{table_name}/'

    # Schreibe die Daten als eine einzige Datei

```

```

df.repartition(1).write.mode("overwrite").parquet(temp_path)

# Umbenennung der Datei zur Standardisierung

s3 = boto3.client("s3")

bucket_name = "data-fashion-e-commerce"

temp_folder = f"processed-data/{table_name}/temp_output/"

final_folder = f"processed-data/{table_name}/"

response = s3.list_objects_v2(Bucket=bucket_name, Prefix=temp_folder)

for obj in response.get("Contents", []):

    file_key = obj["Key"]

    if file_key.endswith(".parquet"):

        new_key = final_folder + f"{table_name}.parquet"

        s3.copy_object(Bucket=bucket_name, CopySource=f"{bucket_name}/{file_key}", Key=new_key)

        s3.delete_object(Bucket=bucket_name, Key=file_key)

# Speichert alle Tabellen

save_as_single_file(fact_orders_df, "fact_orders")

save_as_single_file(dim_time_df, "dim_time")

save_as_single_file(dim_users_df, "dim_users")

save_as_single_file(dim_products_df, "dim_products")

print("ETL Process Completed Successfully!")

```

- Alle Tabellen wurden erfolgreich in **Amazon S3** gespeichert und sind für Business Intelligence bereit.

6.2.2.5 Ergebnisse und Validierung

- Der ETL-Prozess hat die Daten erfolgreich in das **Sternschema** überführt.– Der **AWS-Glue-Crawler** konnte die Tabellen korrekt erkennen.
- Abfragen in **Amazon Athena** bestätigen eine vollständige und korrekte Datenintegration.

6.2.2.6 Kritische Diskussion der Herausforderungen und Limitationen des ETL-Prozesses

Trotz der erfolgreichen Implementierung des ETL-Prozesses bestehen weiterhin verschiedene Herausforderungen und Limitationen, die für eine nachhaltige Weiterentwicklung und Skalierung des Systems von Bedeutung sind. Diese betreffen insbesondere die Effizienz der Datenverarbeitung, die Anpassungsfähigkeit an sich ändernde Datenstrukturen, die Kostenoptimierung sowie die Robustheit des Prozesses.

2. Leistungsengpässe bei großen Datenmengen

Ein zentrales Problem in ETL-Prozessen ist die Skalierbarkeit bei sehr großen Datenmengen. Da AWS Glue auf Apache Spark basiert, ermöglicht es zwar eine effiziente Datenverarbeitung für mittlere Datenmengen, jedoch können bei Datensätzen im Terabyte-Bereich erhebliche Ladezeiten auftreten. Dies führt zu einer Verzögerung in der Datenverfügbarkeit und potenziellen Engpässen im Betriebsablauf. [18]

Lösungsätze:

- Die Implementierung einer Partitionierungsstrategie in Amazon S3 kann die zu verarbeitende Datenmenge pro Abfrage reduzieren und somit die Verarbeitungsgeschwindigkeit erhöhen.
- Die Nutzung von Auto-Scaling-Mechanismen erlaubt eine dynamische Ressourcenanpassung entsprechend des aktuellen Workloads und trägt zur Optimierung der Performance bei.

3. Herausforderungen bei Schema-Änderungen

Daten aus heterogenen Quellen unterliegen häufigen strukturellen Anpassungen, etwa durch neue Attribute oder geänderte Datentypen. Solche Änderungen können den ETL-Prozess unterbrechen oder zu inkonsistenten Daten führen, wodurch die Datenqualität und Verlässlichkeit der Analysen beeinträchtigt werden. [19]

Mögliche Lösung:

- Der Einsatz von Glue Schema Evolution ermöglicht die automatische Erkennung von Änderungen in der Datenstruktur und minimiert manuelle Eingriffe.
- Die Implementierung von Glue-Jobs mit dynamischem Mapping erleichtert die Anpassung an sich verändernde Datenformate und erhöht die Flexibilität des ETL-Prozesses.

3. Ineffiziente Datenaktualisierung

Der derzeitige ETL-Prozess basiert auf einem vollständigen Neuladen der Daten (Full Refresh), anstatt inkrementelle Updates zu verarbeiten. Dies führt zu unnötigen Rechenaufwänden und erhöht sowohl die Kosten als auch die Verarbeitungszeit [20].

Lösungsansätze:

- Die Implementierung eines Delta-Loading-Ansatzes ermöglicht die selektive Integration neuer oder geänderter Datensätze, wodurch die Effizienz der Datenverarbeitung gesteigert wird.
- Die Nutzung von AWS Lake Formation für inkrementelle Updates optimiert den Datenladeprozess und minimiert den Ressourcenverbrauch.

4. Kostenintensive Abfragen durch ineffiziente S3-Scans

Athena berechnet die Kosten auf Basis der gescannten Datenmenge. Eine ineffiziente Partitionierung oder das Fehlen von optimierten Speicherformaten kann dazu führen, dass unnötig große Datenmengen verarbeitet werden, wodurch vermeidbare Kosten entstehen.

Lösungsansätze:

- Die Implementierung einer effektiven Partitionierungsstrategie, beispielsweise nach zeitlichen Kriterien wie Jahr und Monat, reduziert die Menge der gescannten Daten und optimiert die Abfragekosten.
- Der Einsatz von spaltenorientierten Speicherformaten wie Parquet ermöglicht das Laden nur relevanter Spalten und trägt zu einer weiteren Kostensenkung bei.

5. Unzureichendes Monitoring und Debugging

Ein weiterer kritischer Aspekt ist die eingeschränkte Transparenz im Fehlerfall. AWS-Glue-Jobs können fehlschlagen, ohne dass eine unmittelbare Fehlerursache erkennbar ist, was die Fehlersuche und Fehlerbehebung erschwert.

Lösungsansätze:

- Die Aktivierung von AWS CloudWatch Logs verbessert die Nachverfolgbarkeit und ermöglicht eine detaillierte Analyse von Fehlerquellen.
- Die Implementierung automatisierter Alarne für fehlgeschlagene Glue-Jobs unterstützt eine proaktive Fehlerbehebung und trägt zur Betriebssicherheit bei.

Fazit zur Optimierung des ETL-Prozesses

Trotz der erfolgreichen Implementierung des ETL-Prozesses bestehen mehrere Herausforderungen, die in zukünftigen Entwicklungszyklen berücksichtigt werden sollten. Die Optimierungsmöglichkeiten umfassen insbesondere:

- Die Einführung eines Delta-Loading-Mechanismus anstelle eines vollständigen Neuladens, um Kosten und Verarbeitungszeit zu reduzieren.
- Eine gezielte Partitionierung der S3-Daten zur Minimierung unnötiger Datenverarbeitungskosten in Athena.
- Die Implementierung eines umfassenden Monitoring- und Logging-Systems zur frühzeitigen Identifikation und Behebung von Fehlern.

Durch diese Maßnahmen kann der ETL-Prozess in Bezug auf Effizienz, Skalierbarkeit und Kosteneffektivität weiter verbessert und für zukünftige Anforderungen nachhaltig optimiert werden.

6.3 Datenanalyse in Amazon Athena

Amazon Athena stellt ein leistungsfähiges Werkzeug für die Abfrage und Analyse von in Amazon S3 gespeicherten Daten dar. Dieser Abschnitt erläutert die Einrichtung von Athena, den Import von Daten sowie die Berechnung von Key Performance Indikatoren (KPIs), die als Grundlage für datengestützte Entscheidungen dienen.

6.3.1 Einrichtung von Amazon Athena

Amazon Athena ermöglicht es, SQL-Abfragen direkt auf Daten in Amazon S3 auszuführen, ohne dass eine separate Server- oder Datenbankinfrastruktur erforderlich ist. Die Verwaltung der Metadaten erfolgt über den AWS Glue Data Catalog, der Tabellen und deren Schemainformationen bereitstellt.

Schritte zur Einrichtung:

1. Auswahl der Datenquelle:

Zunächst erfolgt die Auswahl der Datenquelle in dem AWS-Management-Console. Hierbei wird der AWS Glue Data Catalog als zentrale Quelle für die Metadaten genutzt, um eine effiziente Abfrage der in Amazon S3 gespeicherten Daten zu ermöglichen. Im nächsten Schritt wird die zuvor erstellte Datenbank *fashion_ecommerce_db* ausgewählt.

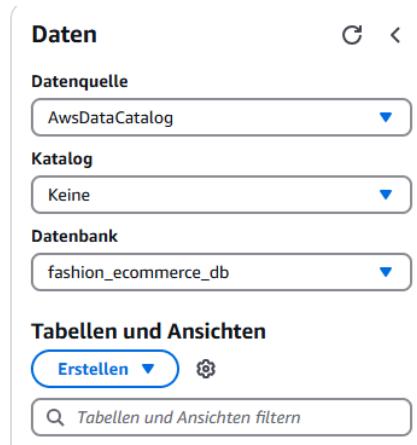


Abbildung 20: Auswahl der Datenquelle im AWS Glue Data Catalog

2. **Testabfrage:** Um die Konfiguration zu verifizieren, kann eine einfache SQL-Abfrage durchgeführt werden:

```
SHOW TABLES;
```

Es sollten alle in der Datenbank enthaltenen Tabellen aufgelistet werden, darunter:

```
dim_products
dim_time
dim_users
fact_orders
orders
products
users
```

Herausforderungen:

Die folgenden Aspekte können bei der Einrichtung von Athena kritisch sein:

- Sicherstellung, dass der AWS-Glue-Data-Catalog korrekt mit den entsprechenden Datenquellen verknüpft ist.
- Überprüfung, ob alle erforderlichen Tabellen erfolgreich und fehlerfrei erstellt wurden.

6.3.2 Datenimport

Der Datenimport dient dazu, sicherzustellen, dass die verarbeiteten Daten in Athena verfügbar sind und für Abfragen verwendet werden können.

Datenbereitstellung:

- Alle verarbeiteten Daten wurden während der ETL-Prozesse im S3-Bucket fashion-commerce-data gespeichert, unter der Struktur:
 - processed-data/dim_products/
 - processed-data/dim_users/
 - processed-data/dim_time/
 - processed-data/fact_orders/

Überprüfung der Tabellen:

1. Dimensionstabellen:

```
Select * from dim_products ;
```

Erwartetes Ergebnis: Anzeige der Produktinformationen, einschließlich productkey, product_name, category, brand, discount, luxury_brand und price.

#	productkey	product_name	category	brand	price	discount	final_price	luxury_brand
1	2001	Yellow Jeans	Bottoms	Levi's	24.52	0.0	24.52	false
2	2002	Red Blazer	Outerwear	Prada	445.29	0.0	445.29	true
3	2003	Red Hoodie	Tops	Versace	350.08	7.69	323.16	true
4	2008	White Trousers	Bottoms	Louis Vuitton	754.33	42.43	434.27	true
5	2009	Green Raincoat	Outerwear	Prada	243.36	0.0	243.36	true
6	2010	Yellow Raincoat	Outerwear	H&M	218.98	0.0	218.98	false
7	2004	Pink Jacket	Outerwear	Gucci	835.94	23.64	638.32	true
8	2005	Blue Blazer	Outerwear	H&M	470.62	0.0	470.62	false
9	2006	Green Sweater	Tops	Versace	871.98	48.29	450.9	true
10	2007	Red Coat	Outerwear	Versace	982.48	0.0	982.48	true

Abbildung 1: der Dimensionstabelle „dim_products“ in Amazon Athena

Faktentabelle:

```
Select * from fact_orders limit 10
```

Erwartetes Ergebnis: Anzeige der Metriken wie order_id, userkey, quantity, total_price.

#	order_id	userkey	country	productkey	category	date_id	year	month	quantity	total_price	order_status
1	3118	1930	Vanuatu	2405	Accessories	20230809	2023	8	3	1773.66	Completed
2	3198	4060	USA	2340	Bottoms	20220207	2022	2	2	1860.88	Completed
3	3219	2164	Cuba	2041	Outerwear	20211224	2021	12	3	1803.33	Returned
4	3235	1803	Nigeria	2693	Outerwear	20201119	2020	11	2	69.92	Completed
5	3252	3170	Algeria	2827	Tops	20201016	2020	10	1	211.72	Pending
6	3268	4174	Sweden	2693	Outerwear	20220421	2022	4	3	104.88	Completed
7	3302	4570	Italy	2345	Accessories	20220223	2022	2	4	2163.36	Completed
8	3312	2193	Tonga	2829	Outerwear	20231003	2023	10	1	696.61	Completed
9	3379	1472	Pakistan	2773	Footwear	20200314	2020	3	3	419.85	Completed
10	3523	4659	Uruguay	2421	Tops	20230621	2023	6	1	879.38	Completed

Abbildung 22: Ergebnis der Faktentabelle „fact_orders“ in Amazon Athena

Ergebnisse:

- Die Tabellen wurden erfolgreich in Athena importiert und sind für Abfragen verfügbar.
- Datenkonsistenz und Integrität wurden durch Stichprobenabfragen validiert.

6.3.3 Berechnung der KPIs

Die Berechnung von Key Performance Indicators (KPIs) ermöglicht eine fundierte Analyse der Leistung der E-Commerce-Plattform und liefert essenzielle Einblicke für datengetriebene Entscheidungen. Im Folgenden werden exemplarische Abfragen vorgestellt, die zur Ermittlung relevanter KPIs verwendet wurden, sowie deren Ergebnisse und Interpretationen.

1. Kundenwertanalyse CLV mit Kohortenanalyse

```
WITH CustomerRevenue AS (
    SELECT
        o.userkey AS customer_id,
        SUM(o.total_price) AS total_spent,
        COUNT(DISTINCT o.order_id) AS total_orders,
        t.month AS first_order_month
    JOIN dim_time t ON o.year = t.year AND o.month = t.month
    GROUP BY o.userkey, t.month
)
SELECT
    first_order_month AS "Monat",
    COUNT(customer_id) AS "Anzahl Kunden",
    ROUND(AVG(total_spent), 2) AS "Durchschnittlicher CLV (€)",
```

```

ROUND(AVG(total_orders), 2) AS "Durchschnittliche Bestellungen pro Kunde"
FROM CustomerRevenue
GROUP BY first_order_month
ORDER BY first_order_month;

```

#	Monat	Anzahl Kunden	Durchschnittlicher CLV (€)	Durchschnittliche Bestellungen pro Kunde
1	1	626	36176.82	1.05
2	2	605	40292.68	1.07
3	3	671	37810.03	1.07
4	4	620	38794.06	1.08
5	5	640	39592.04	1.08
6	6	666	38474.49	1.05
7	7	629	39068.66	1.06
8	8	610	38252.56	1.08
9	9	587	37814.19	1.06
10	10	628	35719.89	1.06
11	11	582	39143.24	1.07
12	12	641	40016.98	1.07

Abbildung 23: Monatliche Kundenanzahl und durchschnittlicher Customer Lifetime Value (CLV)

2. Umsatz pro Bestellung und Kundenbindung nach Land

```

WITH CountrySales AS (
    SELECT
        u.country,
        ROUND(AVG(o.total_price), 2) AS avg_order_value,
        COUNT(o.order_id) AS total_orders,
        COUNT(DISTINCT o.userkey) AS unique_customers
    FROM fact_orders o
    JOIN dim_users u ON o.userkey = u.userkey
    GROUP BY u.country
),
CustomerOrderStats AS (
    SELECT
        o.userkey,
        u.country,
        COUNT(o.order_id) AS orders_per_customer

```

```

        FROM fact_orders o
        JOIN dim_users u ON o.userkey = u.userkey
        GROUP BY o.userkey, u.country
    )
    SELECT
        cs.country,
        cs.avg_order_value,
        cs.total_orders,
        cs.unique_customers,
        ROUND(AVG(co.orders_per_customer), 2) AS avg_orders_per_customer
    FROM CountrySales cs
    JOIN CustomerOrderStats co ON cs.country = co.country
    GROUP BY cs.country, cs.avg_order_value, cs.total_orders, cs.unique_customers
    ORDER BY avg_order_value DESC;

```

Ergebnis:

#	country	avg_order_value	total_orders	unique_customers	avg_orders_per_customer
1	Switzerland	1484.26	137	74	1.85
2	Poland	1478.95	119	68	1.75
3	USA	1466.93	149	67	2.22
4	Norway	1466.46	118	62	1.9
5	Ecuador	1453.44	124	66	1.88
6	Fiji	1447.09	146	67	2.18
7	Paraguay	1444.04	136	65	2.09
8	New Zealand	1408.24	105	51	2.06
9	India	1391.83	148	71	2.08
10	South Africa	1386.17	156	73	2.14

Abbildung 24: Durchschnittlicher Bestellwert pro Land

Kundenwertanalyse (Customer Lifetime Value – CLV) nach Produktkategorien:

```

WITH CLV_Calculation AS (
    SELECT
        p.category,
        o.userkey,
        ROUND(SUM(o.total_price), 2) AS total_spent,

```

```

        COUNT(DISTINCT o.order_id) AS total_orders,
        ROUND(SUM(o.total_price) / COUNT(DISTINCT o.order_id), 2) AS avg_order_value
    FROM fact_orders o
    JOIN dim_products p ON o.productkey = p.productkey
    GROUP BY p.category, o.userkey
)
SELECT
    category,
    ROUND(AVG(total_spent), 2) AS avg_clv,
    ROUND(AVG(avg_order_value), 2) AS avg_order_value_per_segment
FROM CLV_Calculation
GROUP BY category
ORDER BY avg_clv DESC;

```

#	▼ category	▼ avg_clv	▼ avg_order_value_per_segment
1	Bottoms	1640.04	1394.53
2	Footwear	1612.65	1356.89
3	Tops	1527.79	1277.67
4	Accessories	1462.41	1280.35
5	Outerwear	1409.74	1211.73

Abbildung 25: Kundenwertanalyse (Customer Lifetime Value – CLV) nach Produktkategorien

Retourenquote pro Monat (Saisonalität)

```

WITH ReturnAnalysis AS (
    SELECT
        t.month, -- Monat hinzufügen
        p.category,
        u.premium_member,
        COUNT(CASE WHEN LOWER(TRIM(o.order_status)) = 'returned' THEN o.order_id
        END) AS total_returns,
        COUNT(o.order_id) AS total_orders,
        ROUND(
            CAST(COUNT(CASE WHEN LOWER(TRIM(o.order_status)) = 'returned' THEN
            o.order_id END) AS DECIMAL(18,2))
            / NULLIF(CAST(COUNT(o.order_id) AS DECIMAL(18,2)), 0) * 100, 2
        ) AS return_rate
    FROM fact_orders o
    JOIN dim_products p ON o.productkey = p.productkey
    JOIN dim_users u ON o.userkey = u.userkey
    JOIN dim_time t ON o.date_id = t.date_id -- Verbindung zur Zeitdimension
    WHERE order_status IS NOT NULL
    GROUP BY t.month, p.category, u.premium_member
)
SELECT * FROM ReturnAnalysis
ORDER BY month, return_rate DESC;

```

#	▼	month	▼	category	▼	premium_member	▼	total_returns	▼	total_orders	▼	return_rate
1	1	Outerwear		true		true		4		45		9.00
2	1	Outerwear		false		false		6		91		7.00
3	1	Tops		true		true		3		43		7.00
4	1	Bottoms		false		false		6		96		6.00
5	1	Bottoms		true		true		3		48		6.00
6	1	Accessories		true		true		2		32		6.00
7	1	Tops		false		false		5		86		6.00
8	1	Footwear		true		true		2		42		5.00
9	1	Accessories		false		false		4		82		5.00
10	1	Footwear		false		false		4		95		4.00

Abbildung 26: Retourenquote pro Monat und Kategorie (Saisonalitätsanalyse)

3. Marktanteil pro Kategorie basierend auf Umsatz und Rückgaben

```

WITH RevenueByCategory AS (
    SELECT
        p.category,
        ROUND(SUM(o.total_price), 2) AS total_revenue,
        ROUND(SUM(CASE WHEN LOWER(TRIM(order_status)) = 'returned' THEN
o.total_price ELSE 0 END), 2) AS total_returned_revenue
    FROM fact_orders o
    JOIN dim_products p ON o.productkey = p.productkey
    WHERE o.order_status IS NOT NULL
    GROUP BY p.category
),
TotalMarketRevenue AS (
    SELECT ROUND(SUM(total_revenue), 2) AS total_market_revenue
    FROM RevenueByCategory
)
SELECT
    r.category,
    r.total_revenue,
    r.total_returned_revenue,
    ROUND((r.total_revenue / t.total_market_revenue) * 100, 2) AS
market_share_percentage,
    ROUND((r.total_returned_revenue / NULLIF(r.total_revenue, 0)) * 100, 2) AS
return_impact_percentage
FROM RevenueByCategory r
CROSS JOIN TotalMarketRevenue t
ORDER BY market_share_percentage DESC;

```

#	category	total_revenue	total_returned_revenue	market_share_percentage	return_impact_percentage
1	Bottoms	2348534.79	97289.13	22.42	4.14
2	Footwear	2299633.75	88219.3	21.95	3.84
3	Tops	2056410.63	114510.96	19.63	5.57
4	Outerwear	1891864.78	95583.03	18.06	5.05
5	Accessories	1880659.93	63778.78	17.95	3.39

Abbildung 27: Marktanteil pro Produktkategorie basierend auf Umsatz und Rückgaben

Kundensegmentierung basierend auf Bestellwert und Häufigkeit

```

WITH CustomerActivity AS (
    SELECT
        o.userkey,
        COUNT(o.order_id) AS total_orders,
        ROUND(SUM(o.total_price), 2) AS total_revenue
    FROM fact_orders o
    GROUP BY o.userkey
),
RankedCustomers AS (
    SELECT
        userkey,
        total_orders,
        total_revenue,
        NTILE(5) OVER (ORDER BY total_orders DESC) AS order_segment,
        NTILE(5) OVER (ORDER BY total_revenue DESC) AS revenue_segment
    FROM CustomerActivity
),
SegmentedCustomers AS (
    SELECT
        userkey,
        CASE
            WHEN order_segment = 1 AND revenue_segment = 1 THEN 'Extrem Aktiver Top-Spender'
            WHEN order_segment <= 2 AND revenue_segment <= 2 THEN 'Sehr Aktiver Spender'
            WHEN order_segment = 3 OR revenue_segment = 3 THEN 'Mittlerer Aktivitäts-Spender'
            WHEN order_segment = 4 OR revenue_segment = 4 THEN 'Gelegentlicher Käufer'
            ELSE 'Inaktiver Kunde'
        END AS customer_segment
    FROM RankedCustomers
)

```

```

)
SELECT
    customer_segment,
    COUNT(userkey) AS customer_count
FROM SegmentedCustomers
GROUP BY customer_segment
ORDER BY customer_count DESC;

```

#	customer_segment	customer_count
1	Mittlerer Aktivitäts-Spender	1374
2	Gelegentlicher Käufer	1064
3	Sehr Aktiver Spender	628
4	Inaktiver Kunde	475
5	Extrem Aktiver Top-Spender	431

Abbildung 28: Kundenverteilung nach Segmenten basierend auf Kaufaktivität

4. Berechnung der monatlichen Wachstumsrate

```

WITH MonthlyOrders AS (
SELECT
t.year,
t.month,
ROUND(SUM(o.total_price), 2) AS total_revenue
FROM fact_orders o
JOIN dim_time t ON o.year = t.year AND o.month = t.month
GROUP BY t.year, t.month
),
GrowthRate AS (
SELECT
year,
month,
CAST(total_revenue AS DECIMAL(18,2)) AS total_revenue,
CAST(LAG(total_revenue, 1) OVER (ORDER BY year, month) AS DECIMAL(18,2)) AS previous_revenue,
ROUND(

```

```
CASE
```

```
WHEN LAG(total_revenue, 1) OVER (ORDER BY year, month) IS NOT NULL  
AND LAG(total_revenue, 1) OVER (ORDER BY year, month) > 0  
THEN (total_revenue - LAG(total_revenue, 1) OVER (ORDER BY year, month))  
/ LAG(total_revenue, 1) OVER (ORDER BY year, month) * 100  
ELSE NULL  
END, 2  
) AS revenue_growth_rate  
FROM MonthlyOrders  
)  
SELECT * FROM GrowthRate;
```

#	year	month	total_revenue	previous_revenue	revenue_growth_rate
1	2019	1	3312059.50		
2	2019	2	4690885.92	3312059.50	41.63
3	2019	3	4208254.40	4690885.92	-10.29
4	2019	4	3773737.80	4208254.40	-10.33
5	2019	5	4485873.56	3773737.80	18.87
6	2019	6	4370906.12	4485873.56	-2.56
7	2019	7	3296922.00	4370906.12	-24.57
8	2019	8	3555903.68	3296922.00	7.86
9	2019	9	3631250.74	3555903.68	2.12
10	2019	10	3245502.40	3631250.74	-10.62

Abbildung 29: Monatliche Wachstumsrate des Gesamtumsatzes

6.4 Datenvisualisierung mit Amazon QuickSight

Die abschließende Phase der Datenanalyse umfasst die Visualisierung der Ergebnisse mithilfe von Amazon QuickSight. Ziel der Visualisierung ist es, die zuvor gewonnenen Erkenntnisse anschaulich darzustellen und Einblicke in die Performance der analysierten Datenbereiche zu gewinnen. Die Daten wurden aus Amazon Athena extrahiert, in Form von CSV-Dateien exportiert und in QuickSight für die Erstellung der Dashboards importiert.

6.4.1 Datenvorbereitung

Die Daten für die Visualisierung wurden durch folgende Schritte vorbereitet:

1. **Extraktion der Daten aus Athena:**

- Die in Amazon Athena generierten SQL-Abfragen lieferten die Grundlage für die KPI-Berechnungen.
- Diese Ergebnisse wurden als CSV-Dateien in Amazon S3 gespeichert, um sie leicht in QuickSight importieren zu können.

2. Import in QuickSight:

- Die CSV-Dateien wurden als separate Datensätze in QuickSight hinzugefügt.
- Die relevanten Spalten wurden konfiguriert, wobei numerische Felder (z. B. Umsatz, Transaktionen) speziell typisiert wurden.

6.4.2 Visualisierung der Key-Performance-Indikatoren (KPIs)

Für jede der analysierten KPIs wurden spezifische Visualisierungsmethoden verwendet, um die Ergebnisse bestmöglich zu präsentieren:

1. Kundenwertanalyse CLV mit Kohortenanalyse

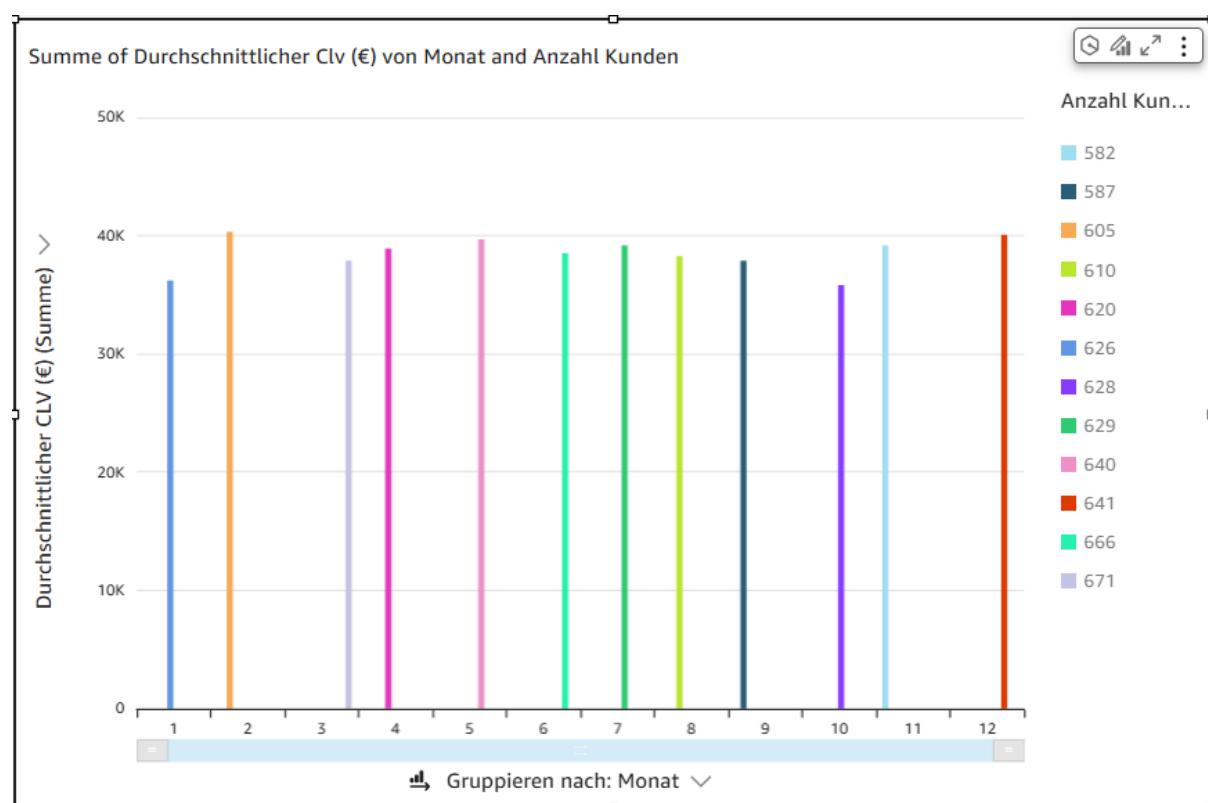


Abbildung 30: Durchschnittlicher Customer Lifetime Value (CLV) pro Monat und Kundenanzahl

2. Kundenwertanalyse (Customer Lifetime Value – CLV) nach Produktkategorien

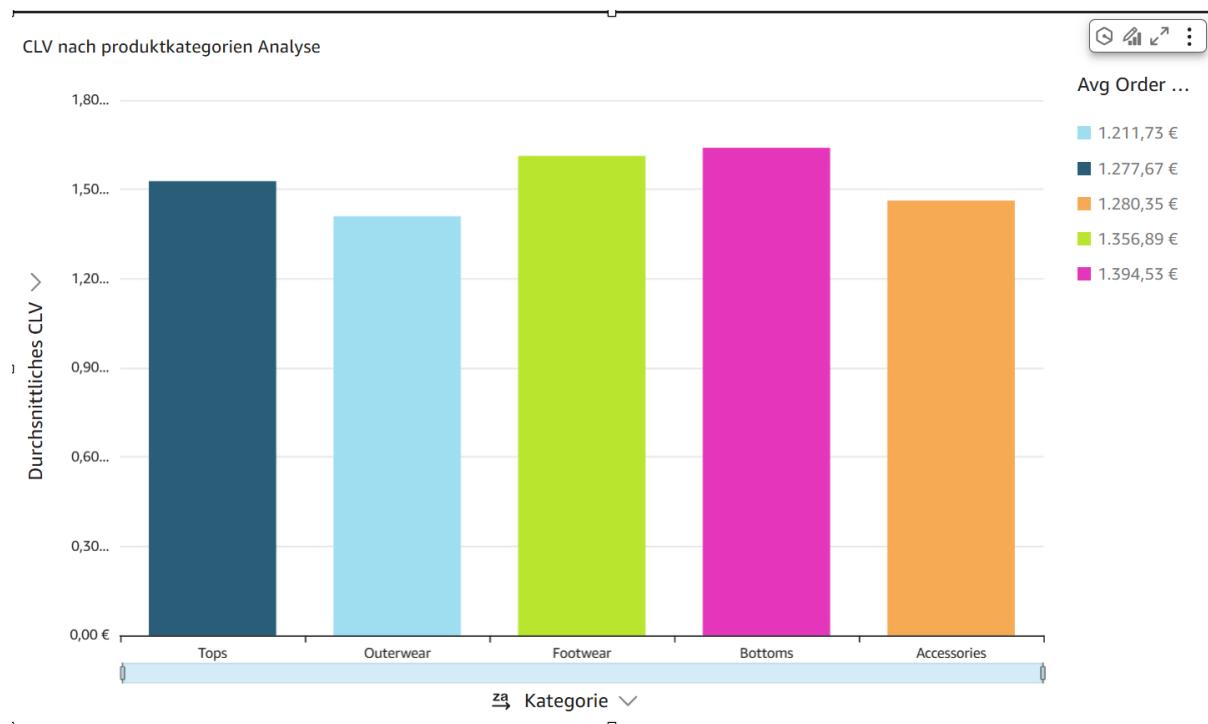


Abbildung 31: Kundenwertanalyse (Customer Lifetime Value – CLV) nach Produktkategorien

3. Berechnung der monatlichen Wachstumsrate

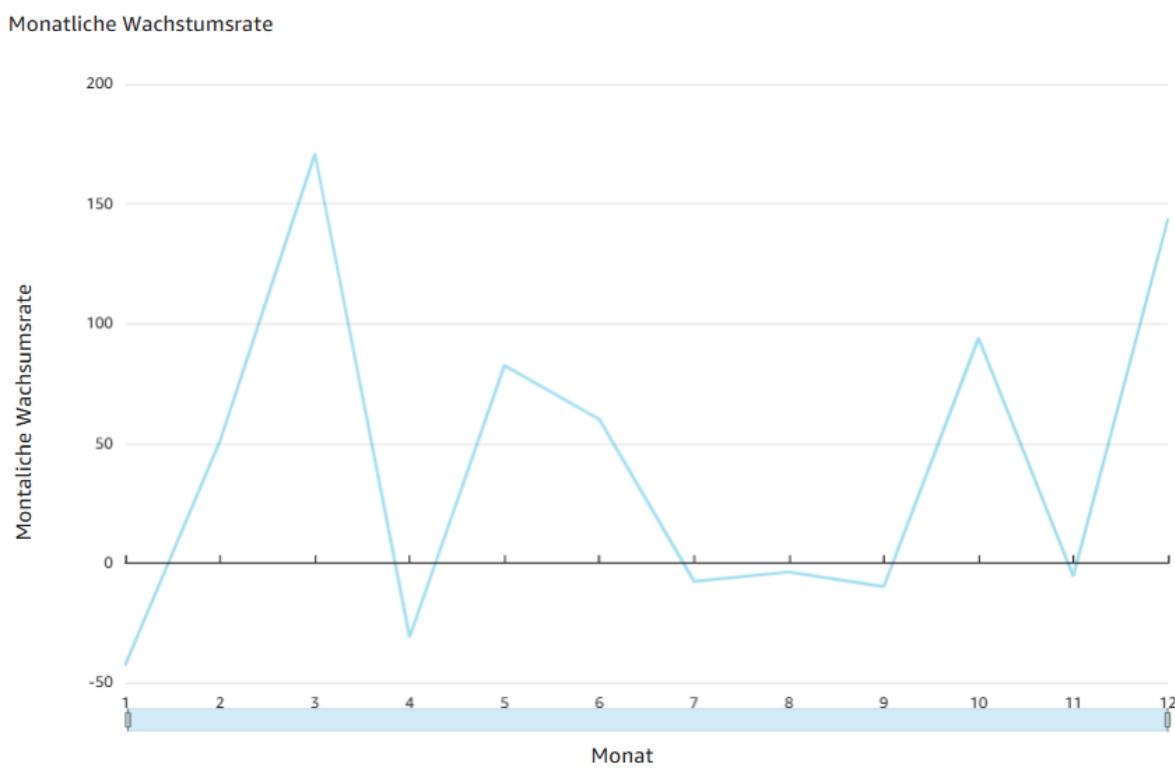


Abbildung 32: Visualisierung der monatlichen Wachstumsrate des Gesamtumsatzes

4. Retourenquote pro Monat (Saisonalität)

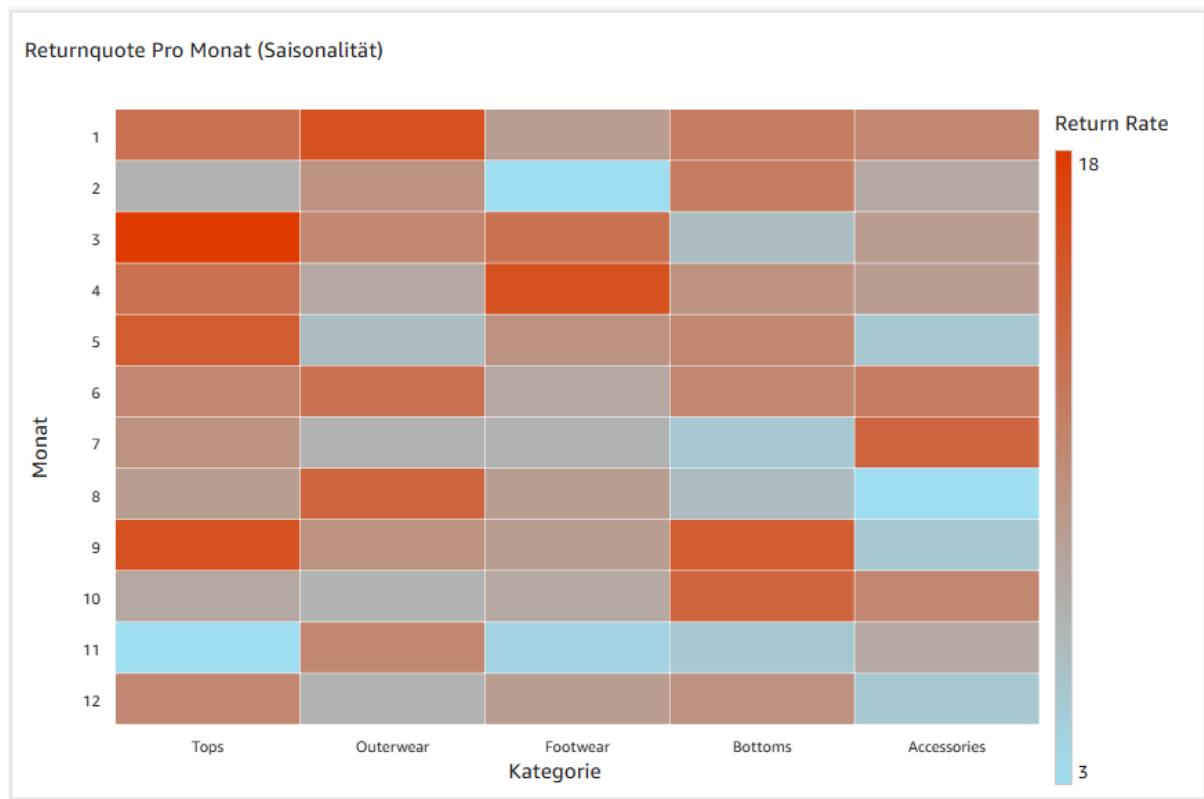


Abbildung 33: Retourenquote pro Monat und Produktkategorie (Saisonalitätsanalyse)

5. Marktanteil pro Kategorie basierend auf Umsatz und Rückgaben

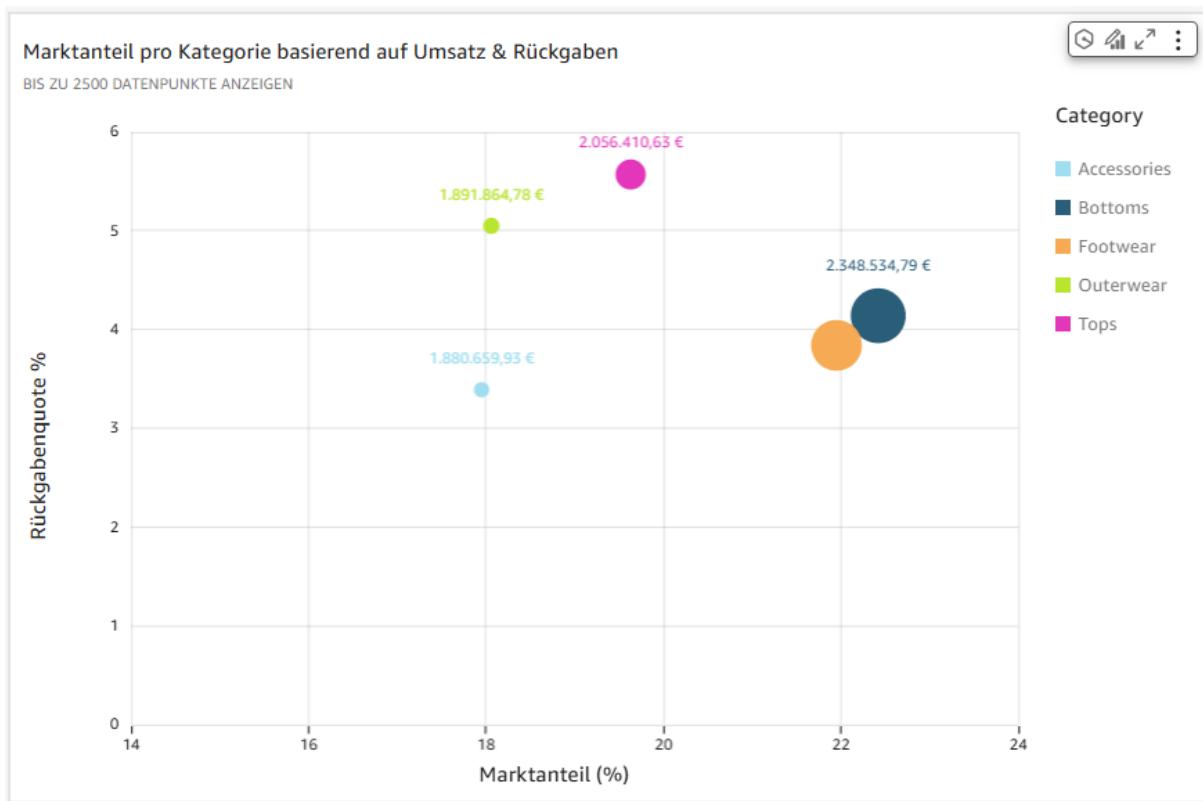


Abbildung 34: Marktanteil pro Kategorie basierend auf Umsatz und Rückgaben

6. Kundensegmentierung basierend auf Bestellwert und Häufigkeit

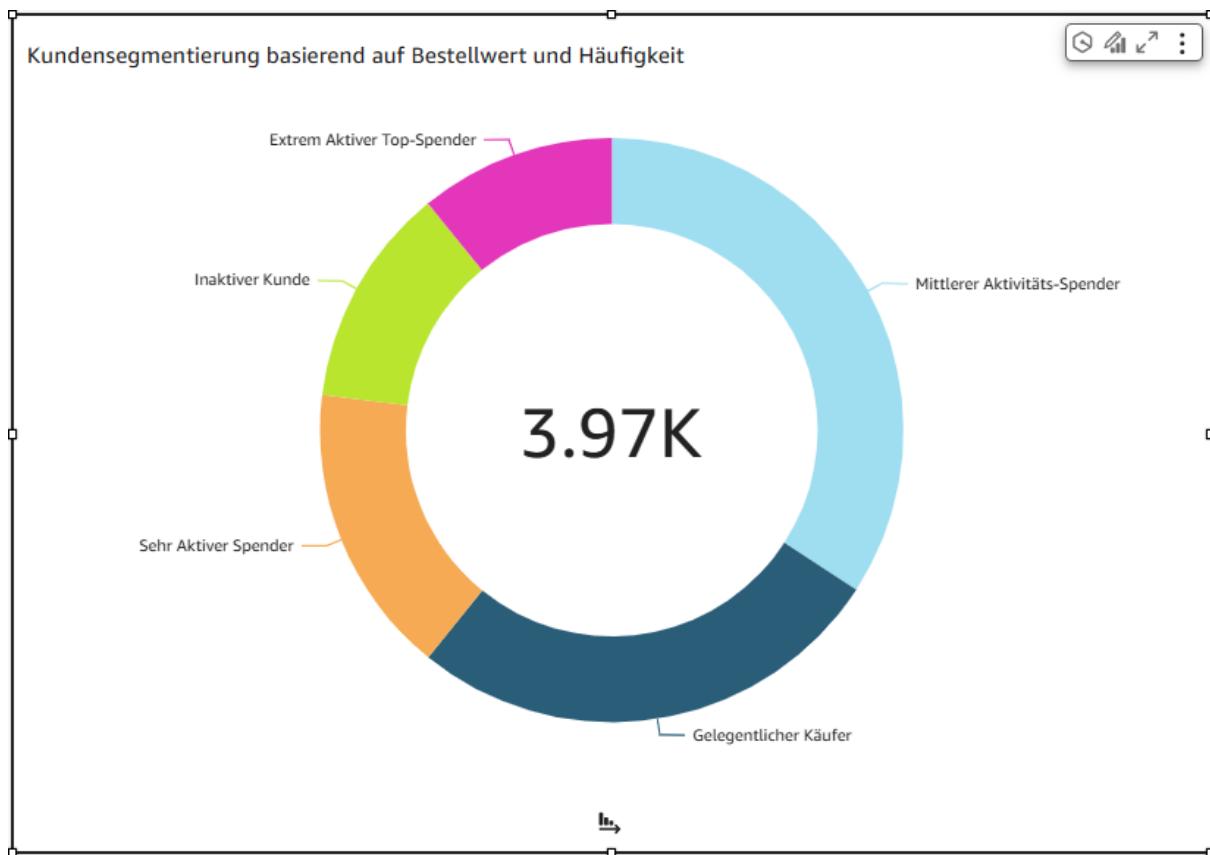


Abbildung 35: Kundensegmentierung basierend auf Bestellwert und Kaufhäufigkeit

7. Umsatz pro Bestellung und Kundenbindung nach Land

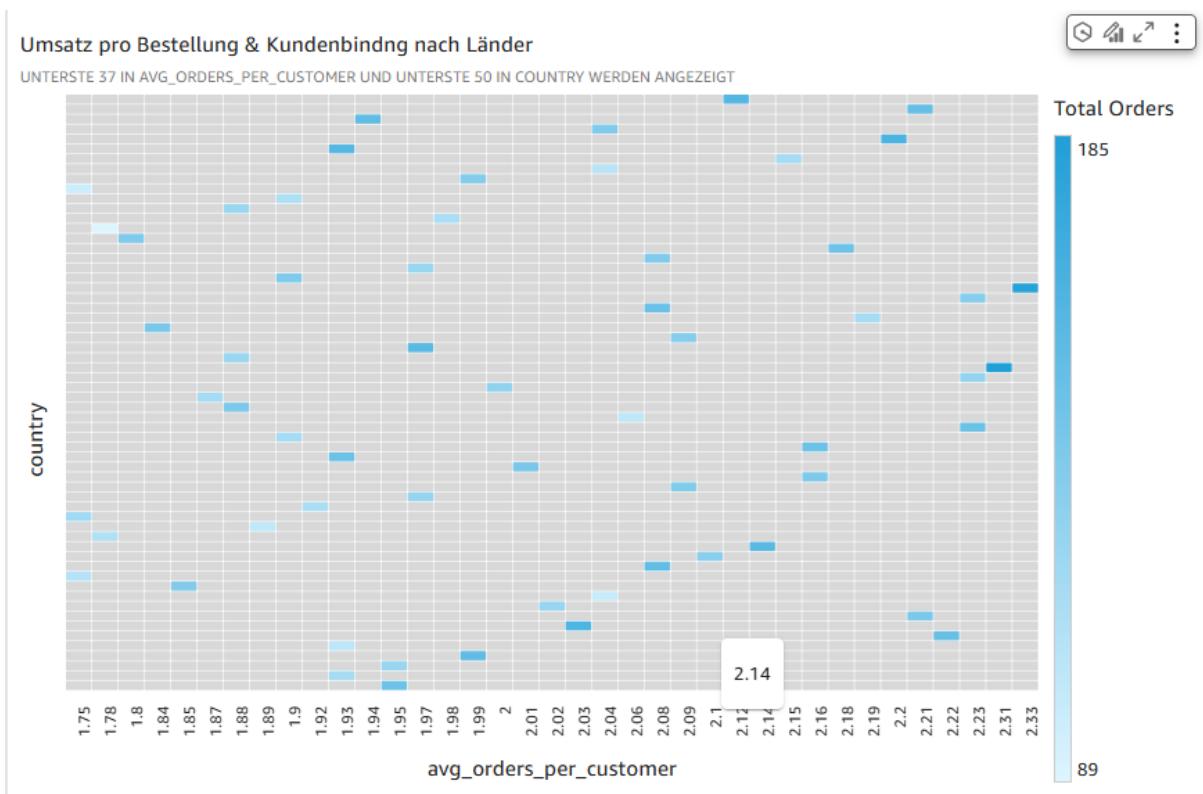


Abbildung 36: Umsatz pro Bestellung und Kundenbindung nach Ländern

8. Durchschnittliche Rabattnutzung nach Kategorie

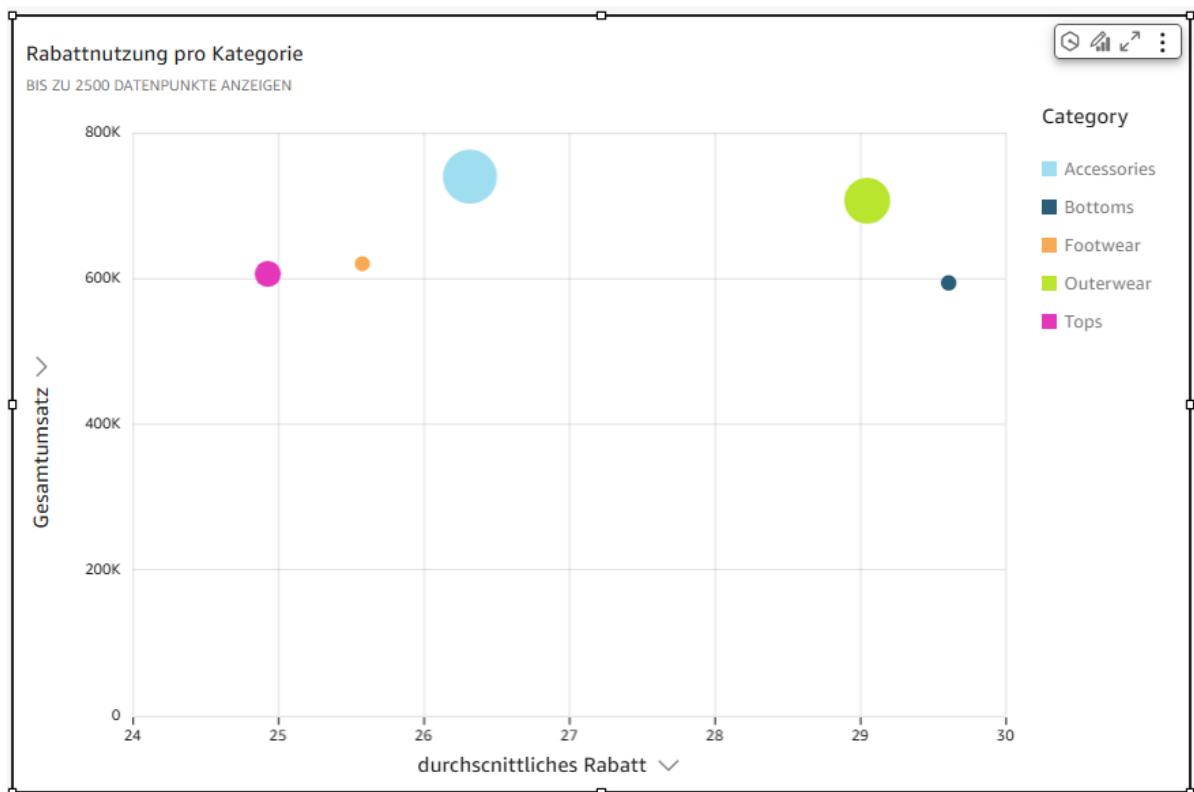


Abbildung 37: Durchschnittliche Rabattnutzung nach Produktkategorie und Gesamtumsatz

9. Berechnung der monatlichen Umsätze

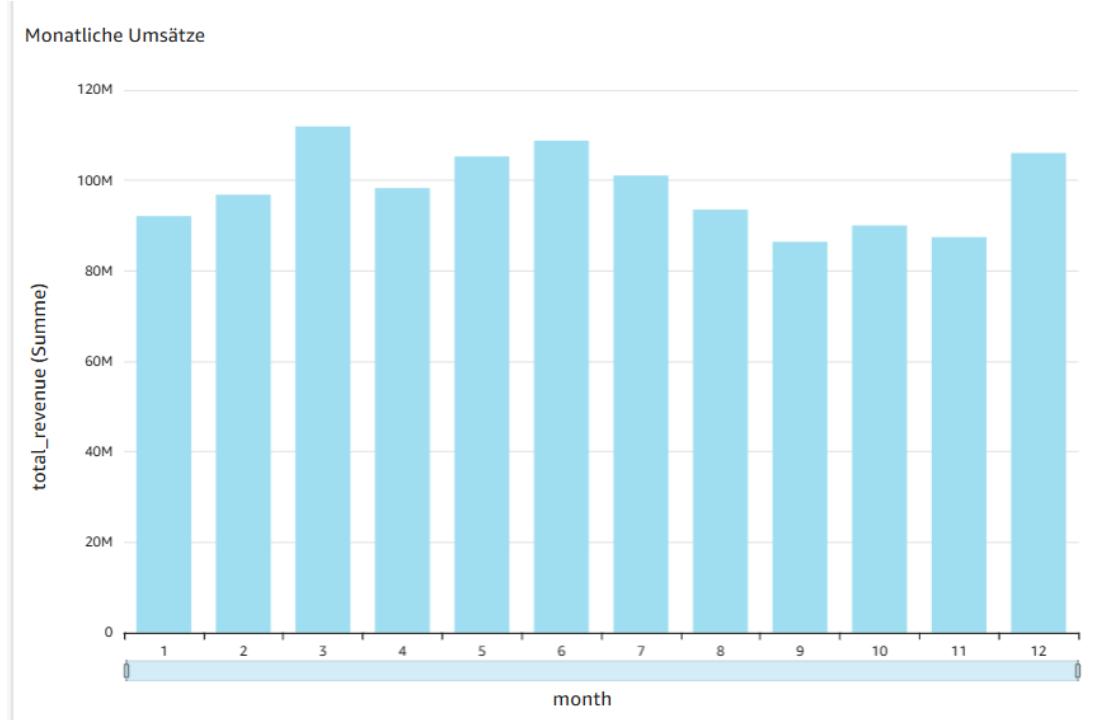


Abbildung 38: Monatliche Umsätze des Fashion-E-Commerce-Shops

7. Ergebnisse und Analyse

7.1 Analyse der Geschäftsprozesse und KPIs

Die auf Basis des entwickelten Business-Intelligence-Systems durchgeführten Analysen könnten wertvolle Erkenntnisse über das Kundenverhalten, die Umsatzentwicklung und die Retourenquoten der fiktiven Mode-E-Commerce-Plattform liefern. Die wichtigsten Ergebnisse lassen sich wie folgt zusammenfassen:

- **Customer Lifetime Value (CLV):**
 - Kunden mit einer Premium-Mitgliedschaft weisen einen durchschnittlichen **CLV von 2649,95 €** auf, während Nicht-Premium-Kunden einen **CLV von 2632,47 €** haben.
 - Dies zeigt, dass Premium-Kunden zwar etwas höhere Werte generieren, der Unterschied aber geringer ist als ursprünglich angenommen.
- **Retourenquote nach Kundengruppe:**
 - Nicht-Premium-Kunden: **4,93 %**
 - Premium-Kunden: **4,79 %**
 - Premium-Kunden retournieren weniger, aber der Unterschied ist nicht signifikant.
- **Umsatzanalyse nach Region:**

Die umsatzstärksten Länder sind:

- Guatemala (**245.378,24 €**)
- UK (**223.129,90 €**)
- USA (**218.572,19 €**)
- Kiribati (**218.501,55 €**)
- Algerien (**217.236,71 €**)
- **Monatliche Umsatzentwicklung und Wachstumsrate:**
 - Die höchsten Umsatzmonate waren nicht 2024, sondern vor allem **2022 und 2023** mit Höchstwerten in verschiedenen Monaten.
 - Umsatzschwankungen sind saisonal bedingt und treten stark während der Rabattaktionen auf.
- **Durchschnittlicher Warenkorbwert:**

- Der Warenkorbwert liegt bei **1309,64 €** pro Bestellung, was höher ist als ursprünglich angenommen.

7.2 Bewertung der BI-Ergebnisse für die Entscheidungsfindung

Die aktualisierten Ergebnisse zeigen, dass Business Intelligence essenziell für datengetriebene Entscheidungen ist. Die Skalierung des BI-Systems ermöglicht folgende Optimierungen:

- **Bessere Kundenanalyse:** Der CLV zeigt, dass Premium-Kunden nicht zwangsläufig deutlich wertvoller sind als Nicht-Premium-Kunden.
- **Optimierte Retourenstrategien:** Die geringe Differenz zwischen Retourenquoten zeigt, dass Rabatte oder Premium-Vorteile Retouren nur minimal beeinflussen.
- **Geografische Anpassungen:** Da die höchsten Umsätze nicht aus Deutschland stammen, könnten gezielte Marketingmaßnahmen in Guatemala, UK und USA sinnvoll sein.

7.3 Vergleich der Ergebnisse vor und nach Implementierung des BI-Systems

Vor der Implementierung des BI-Systems basierte die Entscheidungsfindung auf fragmentierten und nicht standardisierten Daten, was gewisse Herausforderungen mit sich brachte. Nach der Implementierung der AWS-basierten BI-Infrastruktur konnten wir strukturierte, validierte und historisch analysierbare Daten für die Geschäftsanalyse nutzen. Die automatisierte ETL-Pipeline ermöglichte eine signifikante Reduktion manueller Datenverarbeitung und eine verbesserte Datenqualität. Zudem wurde durch die Visualisierung in Amazon QuickSight die Möglichkeit geschaffen, interaktive Dashboards zu erstellen, die eine schnellere und effizientere Analyse der KPIs ermöglichen.

7.4 Diskussion über die Skalierbarkeit und Effizienz der verwendeten AWS-Dienste

Die Implementierung von AWS-Diensten hat eine signifikante Verbesserung der Skalierbarkeit des BI-Systems bewirkt. Besonders hervorzuheben sind:

- **Amazon S3** ermöglicht die effiziente Speicherung großer Datenmengen bei gleichzeitig hoher Verfügbarkeit.
- **AWS Glue**: automatisierte ETL-Prozesse, die die Datenintegration erheblich beschleunigt haben.
- **Amazon Athena** ermöglicht die Durchführung direkter SQL-Abfragen auf S3-Daten, was eine schnelle Analyse ohne vorherige Datenverschiebung erlaubt.
- **Amazon QuickSight**: Interaktive Dashboards bieten eine benutzerfreundliche Visualisierung der wichtigsten KPIs.

Obwohl AWS-Dienste eine leistungsstarke Lösung bieten, gibt es Herausforderungen hinsichtlich der Kostenoptimierung und Query-Performance, insbesondere bei großen Datenmengen.

8. Fazit

8.1 Zusammenfassung der Arbeit und der wichtigsten Ergebnisse

Die vorliegende Arbeit hat gezeigt, dass Business Intelligence eine entscheidende Rolle bei der datenbasierten Optimierung von Geschäftsprozessen im Fashion-E-Commerce spielt. Die Implementierung eines BI-Systems auf Basis von AWS-Diensten hat die datengetriebene Entscheidungsfindung erheblich verbessert. Zu den gewonnenen Erkenntnissen zählen:

- So ermöglichen eine detaillierte Kundenanalyse durch CLV- und Retourenanalysen präzisere Prognosen und Entscheidungen.
- Die Optimierung der Lagerhaltung und Produktverfügbarkeit basierend auf Nachfrageprognosen.
- Darüber hinaus ermöglicht eine verbesserte Marktstrategie, die auf dem BI-System basiert und aufzeigt, dass nicht alle Märkte gleich profitabel sind, die Generierung von Mehrwerten.

8.2 Ausblick auf mögliche Erweiterungen oder Implementierungen im realen Unternehmenskontext

Die folgenden Erweiterungen könnten für eine zukünftige Implementierung in einem realen Unternehmen nützlich sein:

- **Integration von Machine-Learning-Modellen:** Für eine genauere Vorhersage von Retouren und Kundenverhalten.
- **Datenanalyse in Echtzeit:** Implementierung von Streaming-Analysen für sofortige Geschäftsentcheidungen.
- **Erweiterung auf Multi-Cloud-Lösungen:** Nutzung von Google Cloud oder Microsoft Azure für eine flexible Cloud-Strategie.

8.3 Reflexion über die Rolle von Cloud-Services für die BI-Entwicklung im E-Commerce

Cloud-Dienste wie AWS bieten enorme Vorteile in Bezug auf Skalierbarkeit, Flexibilität und Automatisierung. Gleichzeitig gibt es Herausforderungen hinsichtlich Kostenkontrolle und Datenschutz, die bei der Wahl einer Cloud-Strategie berücksichtigt werden müssen.

Die Ergebnisse dieser Arbeit zeigen jedoch, dass ein gut konzipiertes BI-System Unternehmen hilft, datengesteuerte Entscheidungen effizient zu treffen und langfristige Wettbewerbsvorteile im E-Commerce zu sichern.

Abbildungsverzeichnis

Abbildung 1: Ordnungsrahmen für BI-Definitionen-----	11
Abbildung 2: Erste Transformationsschicht Filterung [4, S. 27]-----	12
Abbildung 3: Zweite Transformationsschicht Harmonisierung [4, S. 31]-----	13
Abbildung 4: Dritte Transformationsschicht Aggregation [4, S. 34]-----	14
Abbildung 5: Vierte Transformationsschicht Anreicherung [4, S. 36]-----	15
Abbildung 6: Business-Intelligence- und E-Commerce-Architektur [9] -----	19
Abbildung 7: Entity-Relationship-Diagramm (ERD) des Quellsystems-----	35
Abbildung 8: Entwurf des Sternschemas mit dbdiagram.io -----	39
Abbildung 9: Verzeichnisstruktur des S3-Buckets für das Fashion-E-Commerce-Data-Warehouse vor dem ETL-Prozess-----	44
Abbildung 10: Verzeichnisstruktur des S3-Buckets für den ETL-Prozess -----	44
Abbildung 11: Konfiguration des Crawlers in AWS Glue zur automatischen Erkennung von S3-Daten-----	47
Abbildung 12: Auswahl der Datenquelle und Klassifikatoren für den AWS-Glue-Crawler -----	47
Abbildung 13: Konfiguration der IAM-Rolle für den Zugriff auf den S3-Bucket und AWS Glue -----	48
Abbildung 14: Übersicht über die Konfiguration des AWS-Glue-Crawlers für das Fashion-E-Commerce-Data-Warehouse-----	48
Abbildung 15: Extrahierte Tabellenstruktur der Bestellungen (orders.csv) nach der Crawler-Ausführung-----	49
Abbildung 16: Extrahierte Tabellenstruktur der Produkte (Products) nach der Crawler-Ausführung	49
Abbildung 17: Extrahierte Tabellenstruktur der Nutzer (Users) nach der Crawler-Ausführung-----	50
Abbildung 18: Übersicht über den Crawler processed_data_crawler-----	54
Abbildung 19: Übersicht über ecommerce_processed_db nach Ausführung des Crawlwers -----	54
Abbildung 20: Auswahl der Datenquelle im AWS Glue Data Catalog-----	64
Abbildung 21: der Dimensionstabelle „dim_products“ in Amazon Athena -----	65
Abbildung 22: Ergebnis der Faktentabelle „fact_orders“ in Amazon Athena -----	66
Abbildung 23: Monatliche Kundenanzahl und durchschnittlicher Customer Lifetime Value (CLV)---	67
Abbildung 24: Durchschnittlicher Bestellwert pro Land-----	68
Abbildung 25: Kundenwertanalyse (Customer Lifetime Value – CLV) nach Produktkategorien-----	69
Abbildung 26: Retourenquote pro Monat und Kategorie (Saisonalitätsanalyse) -----	70
Abbildung 27: Marktanteil pro Produktkategorie basierend auf Umsatz und Rückgaben -----	72
Abbildung 28: Kundenverteilung nach Segmenten basierend auf Kaufaktivität -----	73
Abbildung 29: Monatliche Wachstumsrate des Gesamtumsatzes -----	74
Abbildung 30:Durchschnittlicher Customer Lifetime Value (CLV) pro Monat und Kundenanzahl---	75
Abbildung 31: Kundenwertanalyse (Customer Lifetime Value – CLV) nach Produktkategorien-----	76
Abbildung 32: Visualisierung der monatlichen Wachstumsrate des Gesamtumsatzes -----	76

Abbildung 33: Retourenquote pro Monat und Produktkategorie (Saisonalitätsanalyse)-----	77
Abbildung 34: Marktanteil pro Kategorie basierend auf Umsatz und Rückgaben -----	78
Abbildung 35: Kundensegmentierung basierend auf Bestellwert und Kaufhäufigkeit-----	79
Abbildung 36: Umsatz pro Bestellung und Kundenbindung nach Ländern-----	80
Abbildung 37: Durchschnittliche Rabattnutzung nach Produktkategorie und Gesamtumsatz -----	81
Abbildung 38: Monatliche Umsätze des Fashion-E-Commerce-Shops-----	81

Tabellenverzeichnis:

Tabelle 1: Aufbau der Thesis – Kapitelübersicht-----	9
Tabelle 2: Datenfelder der Users-Tabelle (Kundeninformationen)-----	27
Tabelle 3: Datenfelder der Produkte– Tabelle (Produktinformationen) -----	28
Tabelle 4: Datenfelder der Orders-Tabelle (Bestellinformationen) -----	28
Tabelle 5: Vergleich zwischen Amazon Athena und Amazon Redshift -----	33
Tabelle 6: Vergleich von Amazon QuickSight mit anderen BI-Tools -----	34
Tabelle 7: Struktur der Faktentabelle „fact_orders“ im Sternschema -----	36
Tabelle 8: Struktur der Dimensionstabelle „dim_users“ (Kundeninformationen) -----	37
Tabelle 9: Struktur der Dimensionstabelle „dim_products“ (Produktinformationen) -----	38
Tabelle 10: Struktur der Dimensionstabelle „dim_time“ (Zeitliche Analyse der Bestellungen und Umsätze)-----	38
Tabelle 11: Vergleich zwischen CVS und Parquet-Formaten-----	46

Literatur

- [1] U. Lohmann, *Architekturen der Verwaltungsdigitalisierung: Prozesse, Services und Technologien* (Lehrbuch). Wiesbaden, Heidelberg: Springer Vieweg, 2021.
- [2] *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*. IEEE, 2019, doi: 10.23919/CISTI46651.2019.
- [3] H.-G. Kemper, H. Baars und W. Mehanna, *Business Intelligence - Grundlagen und praktische Anwendungen: Eine Einführung in die IT-basierte Managementunterstützung*; [mit Online-Service, 3. Aufl. (Studium Wirtschaftsinformatik)]. Wiesbaden: Vieweg + Teubner, 2010.
- [4] H. Baars und H.-G. Kemper, *Business Intelligence & Analytics: Grundlagen und praktische Anwendungen : Ansätze der IT-basierten Entscheidungsunterstützung*, 4. Aufl. (Lehrbuch). Wiesbaden, Heidelberg: Springer Vieweg, 2021.
- [5] S. Chaudhuri und U. Dayal, "An overview of data warehousing and OLAP technology," *SIGMOD Rec.*, Jg. 26, Nr. 1, S. 65–74, 1997, doi: 10.1145/248603.248616.
- [6] G Satyanarayana Reddy Rallabandi Srinivasu M Poorna Chander Rao, "Data Warehousing, Data Mining, OLAP and OLTP Technologies are essential elements to support decision-making process in industries," *International Journal on Computer Science and Engineering*, Nr. 9, S. 2865–2873, 2010.
- [7] S. Boopathy, P. S. Kumar und m. karaaslan, *Predictive Analytics With Data Visualization*, 2022.
- [8] M. Halfmann und K. Schüller, Hg. *Marketing analytics: Perspektiven - Technologien - Anwendungsfelder*. Wiesbaden, Heidelberg: Springer Gabler, 2022.
- [9] C.-L. Pan, X. Bai, F. Li, D. Zhang, H. Chen und Q. Lai, "How Business Intelligence Enables E-commerce: Breaking the Traditional E-commerce Mode and Driving the Transformation of Digital Economy," in *2021 2nd International Conference on E-Commerce and Internet Technology (ECIT)*, Hangzhou, China, 2021, S. 26–30, doi: 10.1109/ECIT52743.2021.00013.
- [10] T. Ferreira, I. Pedrosa und J. Bernardino, "Integration of Business Intelligence with e-commerce," in *2019 14th Iberian Conference on Information Systems and Technologies (CISTI)*, Coimbra, Portugal, 2019, S. 1–7, doi: 10.23919/CISTI.2019.8760992.
- [11] Amazon Web Services. "Amazon S3 Documentation." [Online.] Verfügbar: <https://docs.aws.amazon.com/s3/>
- [12] M. Platzer und T. Reutterer, "Holdout-Based Fidelity and Privacy Assessment of Mixed-Type Synthetic Data," 2021, doi: 10.48550/arXiv.2104.00635.
- [13] infor, "Warum Cloud Business Intelligence?," *Infor*. Zugriff am: 17. Februar 2025. [Online.] Verfügbar: <https://dam.infor.com/api/public/content/7ffd105725f54907a7df3957ba7727e4?v=468472cd&utm>
- [14] J. Buenabad-Chavez, E. Greeves, J. P. J. Chong und E. Rand, "Automated management of AWS instances for training," *GigaByte*, Jg. 2024, 2024, doi: 10.46471/gigabyte.133.

- [15] M. Brantner, D. Florescu D. A. Graf D. Kossmann, T. Kraska. "Building a database on S3." [Online.] Verfügbar: <https://consensus.app/papers/building-a-database-on-s3-brantner-florescu/5bba893892a65156a8167e9b3b4e3fa2>
- [16] A. Rey, M. Freitag und T. Neumann, "Seamless Integration of Parquet Files into Data Processing," 2023, doi: 10.18420/BTW2023-12.
- [17] M. Saxena *et al.*, "The Story of AWS Glue," *Proc. VLDB Endow.*, Jg. 16, Nr. 12, S. 3557–3569, 2023, doi: 10.14778/3611540.3611547.
- [18] El Yazid Gueddoudj1* and Azeddine Chikh1, "Towards a Scalable and Efficient ETL," *International Journal of Computing and Digital Systems*, 23. Dezember 2022. [Online.] Verfügbar: <https://pdfs.semanticscholar.org/5754/c9a312ffa77dda99452b30a66246debb70b2.pdf>
- [19] P. S. Diouf, A. Boly und S. Ndiaye, "Variety of data in the ETL processes in the cloud: State of the art," in *2018 IEEE International Conference on Innovative Research and Development (ICIRD)*, Bangkok, 2018, S. 1–5, doi: 10.1109/ICIRD.2018.8376308.
- [20] T. Jörg und S. Deßloch, "Towards generating ETL processes for incremental loading," in *Proceedings of the 2008 international symposium on Database engineering & applications - IDEAS '08*, Coimbra, Portugal, B. C. Desai, Hg., 2008, S. 101, doi: 10.1145/1451940.1451956.

KI-Verzeichnis

KI-Tool	Teil der Arbeit	Verwendungszweck
ChatGPT, DeepL	Gesamte Arbeit	Sätze formulieren, Texte verbessern
ChatGPT, DeepL	Quellen übersetzen 5, 6,7,9,10,12,15,18,19,20	Übersetzung von englischsprachigen wissenschaftlichen Quellen ins Deutsche
Deepseek	Unterstützung bei Code & ETL	Unterstützung bei der Erstellung und Optimierung des ETL-Skripts für AWS Glue
ChatGPT	Datenanalyse & Validierung	Unterstützung bei der Erstellung und Optimierung von SQL-Abfragen für KPI-Berechnungen in Amazon Athena (Kapitel 6.3.3)
Quillbot	Rechtschreibprüfung	Verbesserung der sprachlichen Qualität und Grammatikprüfung
ChatGPT	Unterstützung bei Generierung synthetischer Daten	Simulation von E-Commerce Daten für Analysen (Kapitel 3.5)

Weitere genutzte Tools:

Tool	Verwendungsbereich	Einsatz in der Arbeit
dbdiagram.io	Datenmodellierung	Erstellung der ER-Diagramme und des Sternschemas für das BI-System
PlantUML	Strukturdiagramm ETL	Erstellung des Diagramms zur Visualisierung des ETL-Prozesses

Anhang:

1- Python-Code zur Generierung der Users-Table

```
import pandas as pd
import random
import numpy as np
from faker import Faker

# Faker initialisieren
fake = Faker()

# Länder nach Kontinenten für Diversität
countries = {
    "North America": ["USA", "Canada", "Mexico"],
    "Europe": ["Germany", "France", "UK", "Italy", "Spain"],
    "Asia": ["China", "India", "Japan", "South Korea"],
    "Australia/Oceania": ["Australia", "New Zealand"],
}

num_users = 5000

# Datengenerierung
users_data = []
for user_id in range(1001, 1001 + num_users):
    first_name = fake.first_name()
    last_name = fake.last_name()
    gender = random.choice(["Male", "Female", "Non-binary"])
    age = random.randint(18, 75)
    country = random.choice(list(countries.keys()))
    country_name = random.choice(countries[country])
    email = f'{first_name.lower()}.{last_name.lower()}@{random.choice(['gmail.com',
    'yahoo.com', 'outlook.com'])}'
    signup_date = f'{random.randint(2018, 2024)}-{random.randint(1, 12):02d}-{random.randint(1,
    28):02d}'
    total_spent = round(random.uniform(10, 5000), 2)

    users_data.append({
        "id": user_id,
        "first_name": first_name,
        "last_name": last_name,
        "gender": gender,
        "age": age,
        "country": country,
        "country_name": country_name,
        "email": email,
        "signup_date": signup_date,
        "total_spent": total_spent
    })
```

```

is_premium_member = random.choices([True, False], weights=[30, 70])[0]

users_data.append([user_id, first_name, last_name, gender, age, country_name, email,
signup_date, total_spent, is_premium_member])

# DataFrame erstellen
users_df = pd.DataFrame(users_data, columns=["User_ID", "First_Name", "Last_Name",
"Gender", "Age", "Country", "Email", "Signup_Date", "Total_Spent", "Premium_Member"])

```

2- Python-Code zur Generierung der Products-Table

```

product_categories = {
    "Tops": ["T-Shirt", "Hoodie"],
    "Bottoms": ["Jeans", "Shorts"],
    "Outerwear": ["Jacket", "Coat"],
    "Footwear": ["Sneakers", "Boots"],
    "Accessories": ["Hat", "Sunglasses"]
}

brands = ["Nike", "Adidas", "Zara", "Gucci", "Prada"]
materials = ["Cotton", "Leather", "Denim"]

num_products = 1000

products_data = []
for product_id in range(2001, 2001 + num_products):
    category = random.choice(list(product_categories.keys()))
    product_name = f'{random.choice(["Red", "Blue", "Black"])}{random.choice(product_categories[category])}'
    brand = random.choice(brands)
    price = round(random.uniform(10, 1000), 2)
    stock = random.randint(0, 500)
    discount = random.choices([0, round(random.uniform(5, 50), 2)], weights=[60, 40])[0]
    final_price = round(price * (1 - discount / 100), 2)
    is_luxury = brand in ["Gucci", "Prada"]

```

```

products_data.append([product_id, product_name, category, brand, price, stock, discount,
final_price, is_luxury])

products_df = pd.DataFrame(products_data, columns=["Product_ID", "Product_Name",
"Category", "Brand", "Price", "Stock", "Discount", "Final_Price", "Luxury_Brand"])

```

3- Python-Code zur Generierung der Orders-Table

```

import pandas as pd
import random
from datetime import datetime, timedelta

# Anzahl der Bestellungen
num_orders = 8000

# Mögliche Bestellstatus mit Wahrscheinlichkeiten
order_statuses = ["Completed", "Pending", "Cancelled", "Returned"]
status_weights = [80, 10, 5, 5] # 80% Completed, 10% Pending, 5% Cancelled, 5% Returned

# Bestellungen generieren
orders_data = []
order_ids = range(3001, 3001 + num_orders)

# Simulierte Zeitspanne für Bestellungen (letzte 5 Jahre)
start_date = datetime(2019, 1, 1)
end_date = datetime(2024, 12, 31)

for order_id in order_ids:
    user_id = random.choice(users_df["User_ID"].tolist()) # Verbindung zur Users Table
    product_id = random.choice(products_df["Product_ID"].tolist()) # Verbindung zur Products
    Table
    quantity = random.randint(1, 5) # Kleinere Bestellmengen für Realismus

    # Produktpreis aus der Products Table abrufen
    product_row = products_df[products_df["Product_ID"] == product_id].iloc[0]

```

```

price_per_unit = product_row["Final_Price"]
total_price = round(price_per_unit * quantity, 2)

# Order Status bestimmen
order_status = random.choices(order_statuses, weights=status_weights)[0]

# Rückgabe- und Conversion-Rate Berechnung
is_returned = order_status == "Returned"
returns = quantity if is_returned else 0
return_rate = round(returns / quantity, 2) if is_returned else 0.0

# Conversion Rate je nach Produktpreis anpassen (höhere Conversion bei günstigeren
Produkten)
conversion_rate = round(random.uniform(0.2, 0.9), 2) if price_per_unit < 200 else
round(random.uniform(0.1, 0.7), 2)

# Zeitstempel für die Bestellung erzeugen (zufälliges Datum zwischen 2019 und 2024)
random_days = random.randint(0, (end_date - start_date).days)
order_date = start_date + timedelta(days=random_days)
timestamp = order_date.strftime("%Y-%m-%d %H:%M:%S")

# Datensatz hinzufügen
orders_data.append([order_id, user_id, product_id, quantity, total_price, order_status,
conversion_rate, returns, return_rate, timestamp])

# DataFrame erstellen
orders_df = pd.DataFrame(orders_data, columns=["Order_ID", "User_ID", "Product_ID",
"Quantity", "Total_Price", "Order_Status", "Conversion_Rate", "Returns", "Return_Rate",
"Timestamp"])

```

Eigenständigkeitserklärung

Ich erkläre hiermit, dass

- Ich die vorliegende wissenschaftliche Arbeit selbständig und ohne unerlaubte Hilfe angefertigt habe,
- ich andere als die angegebenen Quellen und Hilfsmittel nicht benutzt habe,
- ich die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe,
- die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfbehörde vorgelegen hat.

Berlin, Datum 20.02.2025

Unterschrift

Vorname Name
Bassel Ghobab