

# Open Source Lakehouse Container (mittels DuckDB)

Seminararbeit

vorgelegt am 18. Januar 2025

Fakultät Wirtschaft und Gesundheit

Studiengang Wirtschaftsinformatik

Kurs WWI2022F

von

DAVID KREISMANN

DHBW Stuttgart:

Andreas Buckenhofer  
Dozent für Data Management

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>IV</b>
<b>Tabellenverzeichnis</b>	<b>V</b>
<b>1 Grundlagen</b>	<b>1</b>
1.1 Entstehung des Data Lakehouse . . . . .	1
1.2 Definition und Beschreibung von einem Data Lakehouse . . . . .	2
<b>2 Installation</b>	<b>3</b>
2.1 Werkzeuge . . . . .	3
<b>3 Beispiel</b>	<b>4</b>
3.1 Data Lakehouse . . . . .	4
<b>Literaturverzeichnis</b>	<b>5</b>

# Abkürzungsverzeichnis

<b>ACID</b>	Atomarität, Konsistenz, Isolation und Dauerhaftigkeit
<b>BI</b>	Business Intelligence
<b>DL</b>	Data Lake
<b>DW</b>	Data Warehouse
<b>ETL</b>	Extract, Transform, Load
<b>LH</b>	Data Lakehouse
<b>ML</b>	Machine Learning
<b>OLAP</b>	Online Analytical Processing
<b>OLTP</b>	Online Transaction Processing
<b>SQL</b>	Structured Query Language

# Abbildungsverzeichnis

1	DHBW-Logo 2cm hoch . . . . .	4
---	------------------------------	---

# Tabellenverzeichnis

# 1 Grundlagen

Mit dem Wachstum der Datenmengen setzen Unternehmen zunehmend auf Data-Lakehouse-Architekturen, um strukturierte und unstrukturierte Daten zu verwalten.<sup>1</sup> Weder Data-Lake noch Data-Warehouse-Systeme gelten als ideal für moderne Anwendungsfälle, insbesondere bei fortschrittlichen Analysen wie Machine Learning (ML) Anwendungen, da führende ML-Systeme schlecht auf Data-Warehouses aufbauen.<sup>2</sup> Im Gegensatz zu Business Intelligence (BI)-Abfragen, die kleine Datenmengen verarbeiten, benötigen ML-Systeme große Datensätze und komplexen Code, der über Structured Query Language (SQL) hinausgeht.<sup>3</sup> Dies verdeutlicht die Herausforderungen der aktuellen Datenarchitekturen. Obwohl Cloud-basierte Data-Lake und Data-Warehouse-Lösungen durch die Trennung von Speicher und Rechenressourcen kosteneffizient wirken, führen sie zu erheblicher Komplexität. Moderne Architekturen erfordern oft einen mehrstufigen Extract, Transform, Load (ETL)-Prozess, bei dem Daten zunächst roh im Data Lake und anschließend im Data Warehouse gespeichert werden. Dieser Prozess ist zeitaufwendig, komplex und anfällig für Fehler. Data-Lakehouse-Architekturen lösen diese Probleme, indem sie offene Speicherformate (bspw. Delta Lake) mit Funktionen von Data-Warehouse-Systemen kombinieren, wie ACID-Transaktionen und Abfrageoptimierung (bspw. DuckDB und Ibis).

## 1.1 Entstehung des Data Lakehouse

Traditionelle Datenbanken, sogenannte Online Transaction Processing (OLTP)-Systeme, wurden entwickelt, um tägliche Transaktionen effizient zu verarbeiten und schnellen sowie konsistenten Zugriff auf Daten zu gewährleisten.<sup>4</sup> OLTP-Systeme sind somit optimiert für hohe Transaktionslasten und verwenden normalisierte Datenstrukturen, um Anomalien bei Updates zu vermeiden.<sup>5</sup> Diese starke Normalisierung macht sie jedoch ineffizient für komplexe Analysen, bei denen große Datenmengen verarbeitet oder mehrere Tabellen verknüpft werden müssen.<sup>6</sup> Aus diesem Grund wurden Online Analytical Processing (OLAP)-Systeme entwickelt, die auf Datenanalyse und Entscheidungsunterstützung ausgerichtet sind. OLAP-Queries erfordern oft vollständige Tabellenscans und Aggregationen, wofür OLTP-Systeme ungeeignet sind.<sup>7</sup>

Datenbanken für analytische Zwecke, sogenannte Data Warehouse (DW)s, entstanden als zentrale Speicherorte für strukturierte Daten, die aus verschiedenen Quellen über ETL-Prozesse integriert werden. Diese Daten werden häufig in relationalen Modellen wie Stern- oder Schneeflockenschemata organisiert, um eine effiziente Abfrage und Berichterstellung zu ermöglichen.

---

<sup>1</sup>Vgl. Armbrust u. a. 2021, S. 1

<sup>2</sup>Vgl. Mazumdar/Hughes/Onofre 2023, S. 5

<sup>3</sup>Vgl. Armbrust u. a. 2021, S. 1

<sup>4</sup>Vgl. Vaisman/Zimnyi 2014, S. 45

<sup>5</sup>Vgl. Vaisman/Zimnyi 2014, 45 ff.

<sup>6</sup>Vgl. Vaisman/Zimnyi 2014, 45 ff.

<sup>7</sup>Vgl. Vaisman/Zimnyi 2014, S. 46

DWs sind ideal für Business Intelligence (BI) und historische Analysen, jedoch oft teuer und mit modernen Open-Source- oder Cloud-basierten Tools schwer kompatibel.

Data Lake (DL)s wurden als flexible Alternative zu DWs entwickelt, um große Mengen an unstrukturierten, semi-strukturierten und strukturierten Daten zu speichern. Sie verzichten auf ein vorab definiertes Schema und speichern Daten in ihrer Rohform, was eine hohe Flexibilität bietet. Daten können nach Bedarf organisiert werden, beispielsweise in „Daten-Teiche“ (Data Ponds) für spezifische Datentypen wie rohe Daten, Anwendungsdaten oder Textdaten. Diese Strukturierung erleichtert die Verwaltung, aber DLs stehen vor Herausforderungen wie mangelnder Datenqualität und der Gefahr von „Data Swamps“, in denen unorganisierte und schwer auffindbare Daten die Effektivität einschränken.

## 1.2 Definition und Beschreibung von einem Data Lakehouse

Um die Stärken von DWs und DLs zu vereinen, wurde die Data Lakehouse (LH)-Architektur entwickelt. Sie kombiniert die skalierbare, flexible Speicherfähigkeit von DLs mit den strukturierten und integrierten Analysefunktionen von DWs. LH-Systeme unterstützen fortschrittliche Analysen und ML-Anwendungen und nutzen offene Speicherformate wie Apache Parquet oder Delta Lake sowie Cloud-Objektspeicher wie Minio. Metadatenkataloge wie Apache Iceberg ermöglichen die Verwaltung von Daten und gewährleisten Konsistenz durch ACID-Transaktionen. Performance-Optimierungen wie Indexerstellung, Statistikpflege und effiziente Datenlayouts verbessern zusätzlich die Abfragegeschwindigkeit. Durch ihre Flexibilität, Skalierbarkeit und analytischen Fähigkeiten setzt sich die LH-Architektur zunehmend als Standard in der Datenverarbeitung durch.

## 2 Installation

Bald kann nun der Text Ihrer Projekt- oder Bachelorarbeit beginnen. Dank  $\text{\LaTeX}$  wird Ihre Arbeit garantiert professionell aussehen. Für den Inhalt sind Sie aber weiterhin selbst verantwortlich ;-)

Natürlich ist es schwer, sich vorzustellen, wie das Dokument aussieht, wenn die Vorlage doch gar keinen Text enthält. Aus diesem Grund wird mit Hilfe des Pakets „blindtext“ so genannter Blindtext erzeugt. Mit dem Befehl `\blinddocument` wird nachfolgend ein ganzes Kapitel sinnfreier Blindtext eingefügt.<sup>8</sup>

In Abschnitt 2.1 werden die benötigten Werkzeuge erklärt, bevor dann die Verwendung der Vorlage beschrieben wird. Abschnitt ?? gibt Hilfestellungen für bestimmte Fehler. In Kapitel ?? finden sich Beispiele, wie Sie Quellen korrekt zitieren können. In Kapitel ?? werden Abbildungen, Tabellen, ein Code-Listing und auch mathematische Formeln in den Text eingebunden. Ab Seite 5 finden Sie das Literaturverzeichnis.

### 2.1 Werkzeuge

---

<sup>8</sup>Beachten Sie, dass Sie in Ihrer Arbeit eine Strukturierung wie in Abschnitt 2.1 vermeiden sollten: Dort gibt es einen Abschnitt 2.1.1, aber keinen Abschnitt 2.1.2.



## 3 Beispiel

In diesem Kapitel werden die grundlegenden Aspekte des Themas umfassend analysiert und beschrieben. Zunächst erfolgt eine detaillierte Darstellung des Themas, um ein solides Verständnis zu gewährleisten. Anschließend wird die eigene Umsetzung erläutert und in den Gesamtkontext eingebettet.

### 3.1 Data Lakehouse

In diesem Abschnitt gibt die Abbildungen 1 und ??, die beide das Logo der DHBW zeigen.



Abb. 1: DHBW-Logo 2cm hoch.<sup>9</sup>

---

<sup>9</sup>Mit Änderungen entnommen aus: **OhneAutorenOhneJahr**

# Literaturverzeichnis

- Armbrust, M./Ghodsi, A./Xin, R./Zaharia, M. (2021):** Lakehouse: A New Generation of Open Platforms That Unify Data Warehousing and Advanced Analytics. In.
- Mazumdar, D./Hughes, J./Onofre, J. B. (2023):** The Data Lakehouse: Data Warehousing and More. arXiv: 2310.08697 [cs]. (Abruf: 25.07.2024).
- Vaisman, A./Zimnyi, E. (2014):** Data Warehouse Systems: Design and Implementation. Springer Publishing Company, Incorporated. ISBN: 3-642-54654-4.

# Erklärung zur Verwendung generativer KI-Systeme

Bei der Erstellung der eingereichten Arbeit habe ich die nachfolgend aufgeführten auf künstlicher Intelligenz (KI) basierten Systeme benutzt:

1. Consensus
2. ChatGPT-4o

Ich erkläre, dass ich

- mich aktiv über die Leistungsfähigkeit und Beschränkungen der oben genannten KI-Systeme informiert habe,<sup>10</sup>
- die aus den oben angegebenen KI-Systemen direkt oder sinngemäß übernommenen Passagen gekennzeichnet habe,
- überprüft habe, dass die mithilfe der oben genannten KI-Systeme generierten und von mir übernommenen Inhalte faktisch richtig sind,
- mir bewusst bin, dass ich als Autorin bzw. Autor dieser Arbeit die Verantwortung für die in ihr gemachten Angaben und Aussagen trage.

Die oben genannten KI-Systeme habe ich wie im Folgenden dargestellt eingesetzt:

Arbeitsschritt in der wissenschaftlichen Arbeit	Eingesetzte(s) KI-System(e)	Beschreibung der Verwendungsweise
Literaturrecherche	Consensus	Unterstützung bei der Suche nach wissenschaftlicher Literatur.
Korrektur der Arbeit	ChatGPT-4	Einzelne Kapitel der Arbeit wurden ChatGPT zum Korrigieren gegeben.
Fehleranalyse von Programmcode	ChatGPT-4	Einige Programmzeilen wurden an ChatGPT übergeben, um sie auf Fehler zu prüfen.


---

<sup>10</sup>U.a. gilt es hierbei zu beachten, dass an KI weitergegebene Inhalte ggf. als Trainingsdaten genutzt und wiederverwendet werden. Dies ist insb. für betriebliche Aspekte als kritisch einzustufen.

# Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema: *Open Source Lakehouse Container* (mittels *DuckDB*) selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Stuttgart, 18. Januar 2025  
(Ort, Datum)



(Unterschrift)