Lakehouse mit DuckDB – Konzept und Umsetzung

Seminararbeit

vorgelegt am 19. Januar 2025

Fakultät Wirtschaft und Gesundheit

Studiengang Wirtschaftsinformatik

Kurs WI2022F

von

Raka Pradnya Putra Adita

DHBW Stuttgart:

Andreas Buckenhofer

Inhaltsverzeichnis

Abkürzungsverzeichnis

ACID = Atomicity, Consistency, Isolation, and Durability

CSV = Comma-Separated Values

JSON = JavaScript Object Notation

OLAP = Online Analytical Processing

PII = Personally Identifiable Information

Abbildungsverzeichnis

1 Grundlagen

1.1 Beschreibung und Definition des Lakehouse-Konzepts

Die rapide Zunahme von Datenmengen und die wachsende Bedeutung datengetriebener Analysen haben die Entwicklung neuer Datenarchitekturen vorangetrieben.[1] Traditionelle Datenarchitekturen wie Data Warehouses und Data Lakes verfolgen jeweils unterschiedliche Ansätze und adressieren spezifische Anforderungen.[2]

Data Warehouses konzentrieren sich auf strukturierte Daten und unterstützen analytische Workloads wie OLAP (Online Analytical Processing). Die Implementierung von Schema-on-Write gewährleistet dabei eine hohe Datenqualität und konsistente Analysen.[3] Allerdings sind Data Warehouses oft preisintensiv und weniger flexibel bei der Integration neuer oder unstrukturierter Datenquellen.[4] Sie eignen sich besonders für Reporting und Business Intelligence, zeigen jedoch Schwächen bei der Verarbeitung semi-strukturierter oder unstrukturierter Daten.[5]

Data Lakes wurden entwickelt, um diese Einschränkungen zu überwinden. Sie ermöglichen die Speicherung großer Mengen an Rohdaten in unterschiedlichen Formaten (strukturiert, semi-strukturiert, unstrukturiert). Durch die Flexibilität in der Datenaufnahme und die Verwendung offener Formate wie Apache Parquet oder ORC (Optimized Row Columnar) eignen sich Data Lakes besonders für explorative Analysen und Machine-Learning-Anwendungen.[6] Ohne klare Governance-Richtlinien besteht jedoch die Gefahr, dass Data Lakes zu unübersichtlichen „Data Swamps“ werden, in denen Daten schwer nutzbar sind.[7]

Lakehouses kombinieren die besten Eigenschaften dieser beiden Architekturen.[8] Sie vereinen die Flexibilität und Kosteneffizienz von Data Lakes mit der Datenqualität und den analytischen Fähigkeiten von Data Warehouses. Offene Formate und die Integration von ACID-Transaktionen sorgen für Konsistenz, während sowohl Business Intelligence als auch Machine Learning und Echtzeitanalysen unterstützt werden.[9] Dadurch bieten Lakehouses eine leistungsstarke Lösung für Unternehmen mit komplexen Datenanforderungen.[10]

---

[1] Vgl. Schneider et al. 2024
[2] Vgl. Ebd.
[3] Vgl. Armbrust et al. 2021
[4] Vgl. Schneider, Gröger, Lutsch 2023
[5] Vgl. Ebd.
[6] Vgl. Ebd.
[7] Vgl. Harby, Zulkernine 2022
[8] Vgl. Schneider, Gröger, Lutsch 2023
[9] Vgl. Armbrust et al. 2021
[10] Vgl. Ebd.

1.2 Einführung in DuckDB

DuckDB ist eine moderne eingebettete Datenbank, die speziell für analytische Workloads entwickelt wurde Ihre In-Memory-Architektur und der Verzicht auf einen dedizierten Server machen sie zu einer idealen Lösung für lokale Datenanalysen und Anwendungen im Kontext des Lakehouse-Konzepts.[11]

DuckDB bietet direkten Zugriff auf Daten in offenen Formaten wie Parquet und CSV, ohne dass ein vorheriger Import erforderlich ist. Dadurch können Daten direkt am Speicherort analysiert werden, was komplexe Datenbewegungen und Transformationen reduziert.[12] Die Unterstützung für strukturierte und semi-strukturierte Daten macht DuckDB zu einem flexiblen Werkzeug für moderne Datenarchitekturen.

Die Architektur von DuckDB basiert auf einem columnar-basierten Speichermodell, das Effizienz und Leistung maximiert. Abfragen werden durch eine vektorisierte Query Engine verarbeitet, die Daten in Chunks organisiert. Der Abfrageprozess umfasst mehrere Schritte, darunter Parsing, Optimierung und die physische Ausführung der Abfragen, wie in Abbildung 1 dargestellt. Darüber hinaus kann DuckDB durch die Integration in Programmiersprachen wie Python und R nahtlos in bestehende Workflows eingebettet werden, wodurch es sich problemlos in bestehende Workflows einbinden lässt.

Figure 1.2 A high-level overview of DuckDB's query-processing pipeline

Abb.1: Überblick über die Abfrageverarbeitung in DuckDB[13]

DuckDB bietet mehrere Vorteile, die es von anderen Datenbanksystemen abheben:

- **Einfachheit und Portabilität:** Als eingebettete Datenbank erfordert DuckDB keine Serverinstallation und ist leicht in Skripte oder Anwendungen integrierbar. Dies macht sie ideal für lokale Analysen und Python- oder R-Workflows.
- **Hohe Leistung:** Durch In-Memory-Verarbeitung und moderner Hardware-Unterstützung liefert DuckDB beeindruckende Geschwindigkeit bei analytischen Workloads.[14]

[11] Vgl. Mark Needham, Hunger, Simons 2024
[12] Vgl. Ebd.
[13] Erhalten in: Ebd., S. 3
[14] Vgl. Mark Needham, Hunger, Simons 2024

- **Breite Unterstützung von Formaten:** Native Unterstützung für Formate wie CSV, Parquet und JSON erleichtert die Interoperabilität mit verschiedenen Tools und Pipelines.[15]
- **Kosteneffizient:** Als Open-Source-Lösung verursacht DuckDB keine Lizenzkosten und minimiert durch lokale Verarbeitung Cloud-Ausgaben.

1.3 Medaillon-Architektur

Die Medaillon-Architektur bietet eine skalierbare Designstrategie, um Daten in strukturierten Schichten zu organisieren. Dies fördert Qualität, Verfügbarkeit und Governance. Im Kontext von Lakehouse-Architekturen hilft diese Struktur, die Flexibilität von Data Lakes mit der Datenqualität und den analytischen Fähigkeiten von Data Warehouses zu verbinden.[16][17] Technologien wie DuckDB integrieren sich nahtlos in dieses Konzept.

Die Medaillon-Architektur organisiert Daten in drei Hauptschichten, die jeweils spezifische Funktionen im Datenmanagement übernehmen, wie in Abbildung 2 dargestellt. Die Bronze-Schicht speichert Rohdaten in ihrer ursprünglichen Form und enthält zusätzliche Metadaten wie Zeitstempel und Herkunftsquellen. Sie ermöglicht explorative Analysen und sorgt für die Nachvollziehbarkeit der Datenquellen. Typische Datenformate in dieser Schicht sind JSON, CSV und Parquet.

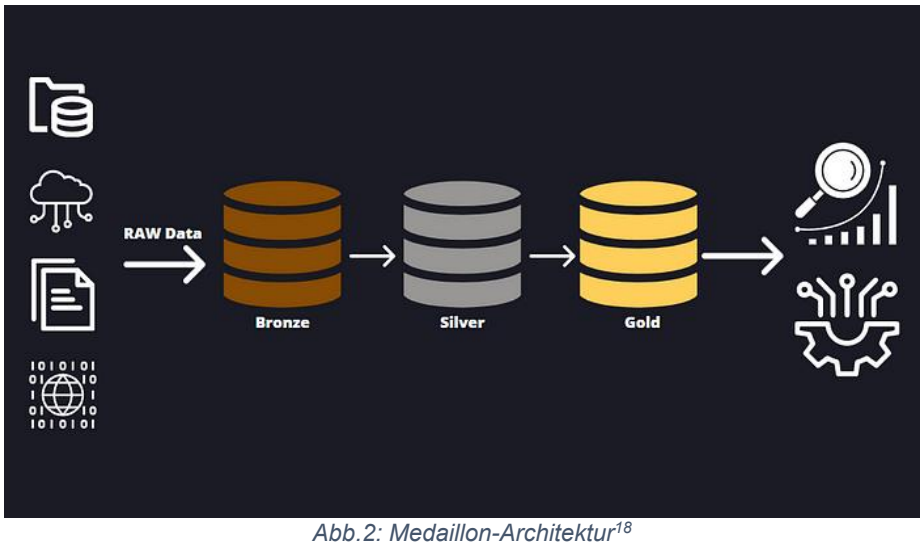

Abb.2: Medaillon-Architektur[18]

Die Bronze-Schicht speichert Rohdaten in ihrer ursprünglichen Form und enthält zusätzliche Metadaten wie Zeitstempel und Herkunftsquellen. Sie ermöglicht explorative Analysen und

[15] Vgl. Ebd.
[16] Vgl. Sirbu, Taleanu, Pop 2024
[17] Vgl. Lopes 2023
[18] Erhalten in: Carvalho 2023

sorgt für die Nachvollziehbarkeit der Datenquellen. [19] Typische Datenformate in dieser Schicht sind JSON, CSV und Parquet.

In der Silber-Schicht werden die Rohdaten aus der Bronze-Schicht transformiert und bereinigt. Fehler werden entfernt, und die Daten werden in ein einheitliches, analysierbares Format gebracht, um qualitativ hochwertige und leicht zugängliche Datensätze sicherzustellen. [20]

Die Gold-Schicht repräsentiert die letzte Verarbeitungsstufe und umfasst aggregierte, geschäftsspezifische Datenmodelle. Diese sind für spezifische Anwendungsfälle wie Dashboards oder Machine-Learning-Anwendungen optimiert und fördern effiziente analytische Workloads. [21]

Die Vorteile der Medaillon-Architektur liegen in der kontinuierlichen Verbesserung der Datenqualität, der klaren Nachvollziehbarkeit von Transformationen sowie der Flexibilität und Wiederherstellbarkeit durch Zugriff auf die unteren Schichten. Darüber hinaus ermöglicht die Architektur granulare Zugriffskontrollen, insbesondere bei sensiblen Daten wie personenbezogenen Informationen (PII). Die Medaillon-Architektur bildet somit die Grundlage für eine methodische und effektive Datenverarbeitung in modernen Lakehouse-Umgebungen.

[19] Vgl. Wiselka o. J.
[20] Vgl. Lopes 2023
[21] Vgl. Wiselka o. J.

2 Installation

In diesem Kapitel werden die detaillierten Schritte zur Installation von DuckDB auf verschiedenen Plattformen sowie den verwendeten Datensatz für das Beispiel beschrieben.

2.1 Installation von DuckDB

DuckDB kann auf verschiedenen Betriebssystemen und Umgebungen installiert werden. Im Folgenden werden die Installationsmethoden für lokale Installationen, die Verwendung von Docker und die Integration als Python-Paket erläutert.

2.1.1 Lokale Installation

macOS:

Für macOS-Benutzer ist die Installation von DuckDB über den Paketmanager Homebrew unkompliziert. Öffnen Sie das Terminal und führen Sie folgenden Befehl aus:

```
brew install duckdb
```

Abb.3: Installation von DuckDB unter macOS

Nach erfolgreicher Installation kann die Installation überprüft werden, indem im Terminal eingegeben wird:

```
duckdb --version
```

Abb.4: Überprüfung der DuckDB-Installation und Version

Windows:

Unter Windows kann DuckDB über den Paketmanager Winget installiert werden. Öffnen Sie die Eingabeaufforderung mit Administratorrechten und führen Sie folgenden Befehl aus:

```
winget install DuckDB.cli
```

Abb.5: Installation von DuckDB unter Windows

Nach der Installation kann die Version überprüft werden mit:

```
duckdb --version
```

Abb.6: Überprüfung der DuckDB-Installation und Version

Linux:

Für Linux-basierte Systeme steht keine Installationsmethode über den Paketmanager zur Verfügung. Alternativ kann DuckDB mit curl heruntergeladen und gebaut werden. Davor muss jedoch die Architektur vom Rechner (x86_64 oder arm64) ausgewählt werden. Öffnen Sie das Terminal und führen Sie folgenden Befehl aus:

```
curl --fail --location --progress-bar --output duckdb_cli-linux-amd64.zip
https://github.com/duckdb/duckdb/releases/download/v1.1.3/duckdb_cli-linux-
amd64.zip && unzip duckdb_cli-linux-amd64.zip
```

Abb.7: Installation von DuckDB unter Linux

Um DuckDB auf dem System zu starten, kann wie folgt vorgegangen werden:

```
duckdb [FILENAME]
```

Abb.8: Typische Starten von DuckDB

Wenn kein [FILENAME]-Argument angegeben wird, öffnet die DuckDB CLI eine temporäre In-Memory-Datenbank. Die Versionsnummer von DuckDB, die Informationen zur Verbindung und eine Eingabeaufforderung, die mit einem `D` beginnt, werden angezeigt.

```
duckdb
```

```
v1.1.3 19864453f7
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent
database.
D
```

Abb.9: Starten von DuckDB für eine In-Memory-Datenbank

Um eine persistente Datenbank zu öffnen oder zu erstellen, kann einfach ein Pfad als Befehlszeilenargument angegeben werden:

```
duckdb my_database.duckdb
```

Abb.10: Starten von DuckDB für eine persistente Datenbank

Weil die Installation auf Linux-System nicht über Paketmanaget ausgeführt wird, wird DuckDB wie folgt gestartet:

```
./duckdb
```

Abb.11: Starten von DuckDB aus dem lokalen Verzeichnis

2.1.2 Installation mit Docker

Es existiert kein offizielles Docker-Image für DuckDB in Docker Hub. Dennoch kann DuckDB in einem Docker-Container genutzt werden, indem ein eigenes Docker-Image erstellt wird. Eine Dockerfile mit folgendem Inhalt kann erstellt werden:

```
FROM ubuntu:latest
RUN apt-get update && apt-get install -y wget unzip
RUN wget https://github.com/duckdb/duckdb/releases/download/v0.3.1/duckdb_cli-
linux-amd64.zip
RUN unzip duckdb_cli-linux-amd64.zip && mv duckdb /usr/local/bin/
ENTRYPOINT ["duckdb"]
```

Abb.12: Dockerfile zur Erstellung eines Docker-Containers mit DuckDB

Das Docker-Image kann gebaut werden mit:

```
docker build -t duckdb_image .
```

Abb.13: Erstellen eines Docker-Images mit DuckDB

Ein Container kann gestartet werden mit:

```
docker run -it --name duckdb_container duckdb_image
```

Abb.14: Starten eines Docker-Containers mit DuckDB

2.1.3 Installation als Python-Paket

DuckDB lässt sich nahtlos in Python integrieren. Das DuckDB-Python-Paket kann mit *pip* installiert werden:

```
pip install duckdb
```

Abb.15: Installation von DuckDB als Python-Paket

Die Installation kann in einer Python-Konsole überprüft werden:

```
import duckdb
print(duckdb.__version__)
```

Abb.16: Überprüfung der installierten DuckDB-Version in Python

Weitere Informationen zur Installation ist unter diesem Link nachzulesen.

2.2 Verwendete Daten

Für die Analyse wird ein E-Commerce-Datensatz verwendet, der das Konsumentenverhalten und Einkaufsgewohnheiten abbildet. Der Datensatz ist auf Kaggle verfügbar. Der *Consumer Behavior and Shopping Habits* Datensatz ist ein umfangreicher Datensatz, der detaillierte Informationen über das Einkaufsverhalten von Verbrauchern im E-Commerce-Bereich enthält.

Hauptmerkmale des Datensatzes:

- **Kategorien**: Der Datensatz umfasst verschiedene Produktkategorien, die es ermöglichen, das Kaufverhalten in unterschiedlichen Segmenten zu untersuchen.
- **Kaufhäufigkeit**: Informationen über die Häufigkeit von Käufen bieten Einblicke in die Wiederkaufraten und die Loyalität der Kunden.
- **Ausgaben pro Transaktion**: Daten über die durchschnittlichen Ausgaben pro Einkauf helfen dabei, das finanzielle Engagement der Verbraucher zu bewerten.

3 Umsetzung eines Beispiels mit DuckDB

Dieses Kapitel beschreibt die praktische Umsetzung eines Lakehouse-Modells mit DuckDB. Bevor mit der praktischen Umsetzung des Lakehouse-Modells begonnen wird, muss DuckDB gestartet werden. Je nach Betriebssystem erfolgt der Start direkt im Terminal oder in der Kommandozeile mit dem Befehl `duckdb`. Alternativ kann eine persistente Datenbank erstellt werden, indem ein Dateiname angegeben wird, zum Beispiel: `duckdb my_database.duckdb`. Nach diesem Schritt werden die Daten in drei Layern verarbeitet: der Bronze Layer für die Rohdatenaufnahme, der Silver Layer zur Datenbereinigung und Transformation sowie der Gold Layer für aggregierte und analytische Daten. Dabei werden SQL-Abfragen und Ergebnisse vorgestellt.

3.1 Bronze Layer: Rohdatenaufnahme

Der Bronze Layer dient dazu, die Rohdaten ohne Modifikationen zu speichern. Die Daten werden aus der Datei `shopping_new.csv` geladen und in eine Tabelle mit dem Namen `Bronze` eingefügt.

```sql
CREATE TABLE Bronze AS
SELECT *
FROM 'shopping_new.csv';
```

Abb.17: SQL-Code für Bronze-Layer

Um SQL-Query auszuführen, kann die Query in Eingabeaufforderung eingegeben werden.

```
D select * from Bronze;
```

Customer ID int64	Age int64	Gender varchar	Item Purchased varchar	Category varchar	Purchase Amount (U… int64	…	Shipping Type varchar	Discount Applied boolean	Promo Code Used boolean	Previous Purchases int64	Payment Method varchar	Frequency of Purch… varchar
1	55	Male	Blouse	Clothing	53	…	Express	true	true	14	Venmo	Fortnightly
2	19	Male	Sweater	Clothing	64	…	Express	true	true	2	Cash	Fortnightly
3	50	Male	Jeans	Clothing	73	…	Free Shipping	true	true	23	Credit Card	Weekly
4	21	Male	Sandals	Footwear	90	…	Next Day Air	true	true	49	PayPal	Weekly
5	45	Male	Blouse	Clothing	49	…	Free Shipping	true	true	31	PayPal	Annually
6	46	Male	Sneakers	Footwear	20	…	Standard	true	true	14	Venmo	Weekly
7	63	Male	Shirt	Clothing	85	…	Free Shipping	true	true	49	Cash	Quarterly
8	27	Male	Shorts	Clothing	34	…	Free Shipping	true	true	19	Credit Card	Weekly
9	26	Male	Coat	Outerwear	97	…	Express	true	true	8	Venmo	Annually
10	57	Male	Handbag	Accessories	31	…	2-Day Shipping	true	true	4	Cash	Quarterly
11	53	Male	Shoes	Footwear	34	…	Store Pickup	true	true	26	Bank Transfer	Bi-Weekly
12	30	Male	Shorts	Clothing	68	…	Store Pickup	true	true	10	Bank Transfer	Fortnightly
13	61	Male	Coat	Outerwear	72	…	Express	true	true	37	Venmo	Fortnightly
14	65	Male	Dress	Clothing	51	…	Express	true	true	31	PayPal	Weekly
15	64	Male	Coat	Outerwear	53	…	Free Shipping	true	true	34	Debit Card	Weekly
16	64	Male	Skirt	Clothing	81	…	Store Pickup	true	true	8	PayPal	Monthly
17	25	Male	Sunglasses	Accessories	36	…	Next Day Air	true	true	44	Debit Card	Bi-Weekly
18	53	Male	Dress	Clothing	38	…	2-Day Shipping	true	true	36	Venmo	Quarterly
19	52	Male	Sweater	Clothing	48	…	Free Shipping	true	true	17	Cash	Weekly
20	66	Male	Pants	Clothing	90	…	Standard	true	true	46	Debit Card	Bi-Weekly
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
3881	42	Female	Shirt	Clothing	20	…	Free Shipping	false	false	40	PayPal	Monthly
3882	56	Female	Scarf	Accessories	25	…	2-Day Shipping	false	false	17	Debit Card	Weekly
3883	35	Female	Pants	Clothing	95	…	Free Shipping	false	false	24	Cash	Fortnightly
3884	34	Female	Hat	Accessories	38	…	Next Day Air	false	false	22	Cash	Fortnightly
3885	47	Female	Sandals	Footwear	29	…	Express	false	false	3	PayPal	Weekly
3886	49	Female	Socks	Clothing	64	…	Free Shipping	false	false	39	Cash	Quarterly
3887	37	Female	Jewelry	Accessories	92	…	Express	false	false	40	Debit Card	Quarterly
3888	40	Female	Shirt	Clothing	34	…	Express	false	false	1	Credit Card	Quarterly
3889	45	Female	Sneakers	Footwear	69	…	Standard	false	false	14	Venmo	Bi-Weekly
3890	57	Female	Dress	Clothing	65	…	Express	false	false	49	Bank Transfer	Annually
3891	35	Female	Shirt	Clothing	81	…	Standard	false	false	33	Debit Card	Annually
3892	36	Female	Dress	Clothing	30	…	Free Shipping	false	false	6	Bank Transfer	Quarterly
3893	35	Female	Jewelry	Accessories	86	…	Standard	false	false	5	PayPal	Fortnightly
3894	21	Female	Hat	Accessories	64	…	Store Pickup	false	false	29	Bank Transfer	Bi-Weekly
3895	66	Female	Skirt	Clothing	78	…	2-Day Shipping	false	false	44	Credit Card	Every 3 Months
3896	40	Female	Hoodie	Clothing	28	…	2-Day Shipping	false	false	32	Venmo	Weekly
3897	52	Female	Backpack	Accessories	49	…	Store Pickup	false	false	41	Bank Transfer	Bi-Weekly
3898	46	Female	Belt	Accessories	33	…	Standard	false	false	24	Venmo	Quarterly
3899	44	Female	Shoes	Footwear	77	…	Express	false	false	24	Venmo	Weekly
3900	52	Female	Handbag	Accessories	81	…	Store Pickup	false	false	33	Venmo	Quarterly

3900 rows (40 shown) 18 columns (12 shown)

Abb.18: Bronze-Tabelle

Beschreibung der Rohdaten:

Die Datei `shopping_new.csv` enthält folgende Attribute:

- **Customer ID**: Eine eindeutige Identifikationsnummer für den Kunden.
- **Age**: Das Alter des Kunden.
- **Gender**: Das Geschlecht des Kunden.
- **Item Purchased**: Der gekaufte Artikel.
- **Category**: Die Kategorie des Artikels.
- **Purchase Amount (USD)**: Der Betrag des Kaufs in US-Dollar.
- **Location**: Der Standort des Kunden.
- **Size, Color, Season**: Eigenschaften des gekauften Artikels.
- **Review Rating**: Bewertung des Produkts durch den Kunden.
- **Subscription Status**: Gibt an, ob der Kunde ein Abonnement hat.
- **Weitere Felder**: Versandart, Rabatte, Zahlungsmethode, etc.

3.2 Silver Layer: Datenbereinigung und Transformation

Im Silver Layer werden die Rohdaten bereinigt und transformiert, um sie für analytische Zwecke vorzubereiten. In diesem Beispiel werden Datentypen konvertiert, fehlende Werte behandelt und neue Attribute erstellt.

```
CREATE TABLE Silver AS
SELECT
    "Customer ID",
    CAST(AGE AS INTEGER) AS Age,
    Gender,
    "Item Purchased",
    Category,
    CAST("Purchase Amount (USD)" AS DECIMAL) AS
PurchaseAmount,
    Location,
    Size,
    Color,
    Season,
    CASE
        WHEN "Review Rating" IS NULL THEN 0
        ELSE CAST("Review Rating" AS INTEGER)
    END AS ReviewRating,
    CASE
        WHEN "Subscription Status" = 'Yes' THEN
TRUE
        ELSE FALSE
    END AS SubscriptionStatus,
    "Shipping Type",
    "Discount Applied",
    "Promo Code Used",
    CAST("Previous Purchases" AS INTEGER) AS
PreviousPurchases,
    "Payment Method",
    "Frequency of Purchases"
FROM Bronze
WHERE "Customer ID" IS NOT NULL AND "Purchase
Amount (USD)" > 0;
```

Abb.19: SQL-Code für Silver Layer

Die Transformationen umfassen folgende Schritte: Das Alter wird in einen Integer-Wert umgewandelt, und der Kaufbetrag wird als Dezimalwert gespeichert. Fehlende Bewertungen werden auf 0 gesetzt. Zudem wird der Abonnementstatus in einen Boolean-Wert umgewandelt, um die Daten zu vereinheitlichen. Kunden ohne ID und Käufe mit einem Betrag von 0 oder weniger werden ausgeschlossen, um die Datenqualität zu gewährleisten.

```
D select * from Silver;
```

Customer ID int64	Age int32	Gender varchar	Item Purchased varchar	Category varchar	PurchaseAmount decimal(18,3)	…	Shipping Type varchar	Discount Applied boolean	Promo Code Used boolean	PreviousPurchases int32	Payment Method varchar	Frequency of Purch… varchar
1	55	Male	Blouse	Clothing	53.000	…	Express	true	true	14	Venmo	Fortnightly
2	19	Male	Sweater	Clothing	64.000	…	Express	true	true	2	Cash	Fortnightly
3	50	Male	Jeans	Clothing	73.000	…	Free Shipping	true	true	23	Credit Card	Weekly
4	21	Male	Sandals	Footwear	90.000	…	Next Day Air	true	true	49	PayPal	Weekly
5	45	Male	Blouse	Clothing	49.000	…	Free Shipping	true	true	31	PayPal	Annually
6	46	Male	Sneakers	Footwear	20.000	…	Standard	true	true	14	Venmo	Weekly
7	63	Male	Shirt	Clothing	85.000	…	Free Shipping	true	true	49	Cash	Quarterly
8	27	Male	Shorts	Clothing	34.000	…	Free Shipping	true	true	19	Credit Card	Weekly
9	26	Male	Coat	Outerwear	97.000	…	Express	true	true	8	Venmo	Annually
10	57	Male	Handbag	Accessories	31.000	…	2-Day Shipping	true	true	4	Cash	Quarterly
11	53	Male	Shoes	Footwear	34.000	…	Store Pickup	true	true	26	Bank Transfer	Bi-Weekly
12	30	Male	Shorts	Clothing	68.000	…	Store Pickup	true	true	10	Bank Transfer	Fortnightly
13	61	Male	Coat	Outerwear	72.000	…	Express	true	true	37	Venmo	Fortnightly
14	65	Male	Dress	Clothing	51.000	…	Express	true	true	31	PayPal	Weekly
15	64	Male	Coat	Outerwear	53.000	…	Free Shipping	true	true	34	Debit Card	Weekly
16	64	Male	Skirt	Clothing	81.000	…	Store Pickup	true	true	8	PayPal	Monthly
17	25	Male	Sunglasses	Accessories	36.000	…	Next Day Air	true	true	44	Debit Card	Bi-Weekly
18	53	Male	Dress	Clothing	38.000	…	2-Day Shipping	true	true	36	Venmo	Quarterly
19	52	Male	Sweater	Clothing	48.000	…	Free Shipping	true	true	17	Cash	Weekly
20	66	Male	Pants	Clothing	90.000	…	Standard	true	true	46	Debit Card	Bi-Weekly
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.
3881	42	Female	Shirt	Clothing	20.000	…	Free Shipping	false	false	40	PayPal	Monthly
3882	56	Female	Scarf	Accessories	25.000	…	2-Day Shipping	false	false	17	Debit Card	Weekly
3883	35	Female	Pants	Clothing	95.000	…	Free Shipping	false	false	24	Cash	Fortnightly
3884	34	Female	Hat	Accessories	38.000	…	Next Day Air	false	false	22	Cash	Fortnightly
3885	47	Female	Sandals	Footwear	29.000	…	Express	false	false	3	PayPal	Weekly
3886	49	Female	Socks	Clothing	64.000	…	Free Shipping	false	false	39	Cash	Quarterly
3887	37	Female	Jewelry	Accessories	92.000	…	Express	false	false	40	Debit Card	Quarterly
3888	40	Female	Shirt	Clothing	34.000	…	Express	false	false	1	Credit Card	Quarterly
3889	45	Female	Sneakers	Footwear	69.000	…	Standard	false	false	14	Venmo	Bi-Weekly
3890	57	Female	Dress	Clothing	65.000	…	Express	false	false	49	Bank Transfer	Annually
3891	35	Female	Shirt	Clothing	81.000	…	Standard	false	false	33	Debit Card	Annually
3892	36	Female	Dress	Clothing	30.000	…	Free Shipping	false	false	6	Bank Transfer	Quarterly
3893	35	Female	Jewelry	Accessories	86.000	…	Standard	false	false	5	PayPal	Fortnightly
3894	21	Female	Hat	Accessories	64.000	…	Store Pickup	false	false	29	Bank Transfer	Bi-Weekly
3895	66	Female	Skirt	Clothing	78.000	…	2-Day Shipping	false	false	44	Credit Card	Every 3 Months
3896	40	Female	Hoodie	Clothing	28.000	…	2-Day Shipping	false	false	32	Venmo	Weekly
3897	52	Female	Backpack	Accessories	49.000	…	Store Pickup	false	false	41	Bank Transfer	Bi-Weekly
3898	46	Female	Belt	Accessories	33.000	…	Standard	false	false	24	Venmo	Quarterly
3899	44	Female	Shoes	Footwear	77.000	…	Express	false	false	24	Venmo	Weekly
3900	52	Female	Handbag	Accessories	81.000	…	Store Pickup	false	false	33	Venmo	Quarterly

3900 rows (40 shown) 18 columns (12 shown)

Abb.20: Silver-Tabelle

3.3 Gold Layer: Aggregierte und analytische Daten

Im Gold Layer werden die bereinigten Daten aus dem Silver Layer weiterverarbeitet, um aggregierte und analytische Informationen bereitzustellen. Dabei werden drei spezifische Analysen durchgeführt: durchschnittliche Kundenausgaben, Top-Produkte sowie ein Vergleich zwischen Abonnenten und Nicht-Abonnenten.

```sql
CREATE TABLE Gold_Customer_Average_Spend AS
SELECT
    "Customer ID",
    AVG(PurchaseAmount) AS AveragePurchaseAmount
FROM Silver
GROUP BY "Customer ID";
```

Abb.21: SQL-Code für Gold-Layer

```sql
CREATE TABLE Gold_Top_Products AS
SELECT
    "Item Purchased",
    Category,
    COUNT(*) AS PurchaseCount
FROM Silver
GROUP BY "Item Purchased", Category
ORDER BY PurchaseCount DESC
LIMIT 5;
```

Abb.22: SQL-Code für Gold-Layer

```sql
CREATE TABLE Gold_Subscription_Comparison AS
SELECT
    SubscriptionStatus,
    AVG(PurchaseAmount) AS AveragePurchaseAmount
FROM Silver
GROUP BY SubscriptionStatus;
```

Abb.23: SQL-Code für Gold-Layer

Die durchschnittlichen Ausgaben pro Kunde werden visualisiert, um das Kaufverhalten verschiedener Kundengruppen zu verstehen, wie in Abbildung 24 dargestellt. Die Top-Produkte und Kategorien werden identifiziert, um Marketingstrategien zu optimieren, wie in Abbildung 25 dargestellt. Zudem wird ein Vergleich zwischen Abonnenten und Nicht-Abonnenten durchgeführt, der aufzeigt, ob Abonnenten im Durchschnitt höhere Ausgaben tätigen, wie in Abbildung 26 dargestellt.

```
D select * from Gold_Customer_Average_Spend;
```

Customer ID int64	AveragePurchaseAmount double
1	53.0
2	64.0
3	73.0
4	90.0
5	49.0
6	20.0
7	85.0
8	34.0
9	97.0
10	31.0
11	34.0
12	68.0
13	72.0
14	51.0
15	53.0
16	81.0
17	36.0
18	38.0
19	48.0
20	90.0
·	·
·	·
·	·
3881	20.0
3882	25.0
3883	95.0
3884	38.0
3885	29.0
3886	64.0
3887	92.0
3888	34.0
3889	69.0
3890	65.0
3891	81.0
3892	30.0
3893	86.0
3894	64.0
3895	78.0
3896	28.0
3897	49.0
3898	33.0
3899	77.0
3900	81.0

3900 rows (40 shown) 2 columns

Abb.24: Durchschnittlicher Kaufbetrag pro Kunde

```
D select * from Gold_Top_Products;
```

Item Purchased varchar	Category varchar	PurchaseCount int64
Pants	Clothing	171
Blouse	Clothing	171
Jewelry	Accessories	171
Shirt	Clothing	169
Dress	Clothing	166

Abb.25: Top 5 Produkten und Kategorien

Die Tabelle `Gold_Top_Products` zeiget, dass Artikel wie Hosen, Blusen und Schmuck die meistgekauften Produkte sind, jeweils mit einer Kaufhäufigkeit von 171. Sie gehören zu den Kategorien "Clothing" und "Accessories". Shirts und Kleider, ebenfalls in der Kategorie "Clothing", folgen mit 169 bzw. 166 Käufen. Dieses Ergebnis verdeutlicht, dass die Kategorie "Clothing" die dominierende Produktkategorie ist.

```
D select * from Gold_Subscription_Comparison;
```

SubscriptionStatus boolean	AveragePurchaseAmount double
false	59.865121180189675
true	59.49192782526116

Abb.26: Vergleich: Abonnenten vs. Nicht-Abonnenten

Die Tabelle `Gold_Subsription_Comparison` zeigt, dass die durchschnittlichen Ausgaben von Nicht-Abonnenten bei etwa 59,87 USD liegen, während Abonnenten im Durchschnitt 59,49 USD ausgeben. Der Unterschied zwischen den beiden Gruppen ist relativ gering, deutet jedoch darauf hin, dass der Abonnementstatus nur einen minimalen Einfluss auf die durchschnittlichen Kaufbeträge hat.

Literaturverzeichnis

Armbrust, Michael et al. 2021. »Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics«.

Carvalho, Rui 2023. *What is Medallion Architecture and How Can You Use It in Fabric?*, https://blog.det.life/what-is-medallion-architecture-and-how-can-you-use-it-in-fabric-6b0056ababc4 (Zugriff vom 18.1.2025).

Harby, Ahmed A.; Zulkernine, Farhana 2022. »From Data Warehouse to Lakehouse: A Comparative Review«, in *2022 IEEE International Conference on Big Data (Big Data)*, S. 389–395. Osaka, Japan: IEEE.

Lopes, Fábio Rafael Santos 2023. *Lakehouse Data Architecture Data as a first-class citizen within an organization*. NOVA Information Management School.

Mark Needham; Hunger, Michael; Simons, Michael 2024. *DuckDB in Action*. Manning Publications Co.

Schneider, Jan et al. 2024. »The Lakehouse: State of the Art on Concepts and Technologies«, in *SN Computer Science* 5, 5, S. 449.

Schneider, Jan; Gröger, Christoph; Lutsch, Arnold 2023. »The Data Platform Evolution: From Data Warehouses over Data Lakes to Lakehouses«.

Sirbu, Daniel-Ilie; Taleanu, Andrei-Traian; Pop, Florin 2024. »Replication as Lineage Mechanism for Materialized Views in Lakehouse Architectures«, in *2024 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, S. 1–7. Craiova, Romania: IEEE.

Wiselka, Michal o. J. . »Development of Modern Data Platform using Medallion Architecture«.

Erklärung zur Verwendung generativer KI-Systeme

Bei der Erstellung der eingereichten Arbeit habe ich auf künstlicher Intelligenz (KI) basierte Systeme benutzt:

☒ ja
☐ nein[22]

Falls ja: Die nachfolgend aufgeführten auf künstlicher Intelligenz (KI) basierten Systeme habe ich bei der Erstellung der eingereichten Arbeit benutzt:

1. ChatGPT
2. Consensus

Ich erkläre, dass ich

- mich aktiv über die Leistungsfähigkeit und Beschränkungen der oben genannten KI-Systeme informiert habe,[23]
- die aus den oben angegebenen KI-Systemen direkt oder sinngemäß übernommenen Passagen gekennzeichnet habe,[24]
- 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[25]	**Eingesetzte(s) KI- System(e)**	**Beschreibung der Verwendungsweise**
Literaturrecherche	Consensus	Suche nach hochwertigen Quellen zu dem Thema
Korrektur der Arbeit	ChatGPT	ChatGPT als Formulierungshilfe verwendet, teilweise Nachbearbeitung durch Veränderung der Bedeutung nötig

19.01.2025
(Ort, Datum)

(Unterschrift)

[22] Die Erklärung ist in jedem Fall zu unterzeichnen, auch wenn Sie keine KI-Systeme genutzt haben und Ihr Kreuz bei „nein“ gesetzt haben.

[23] 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.

[24] In der Fußnote Ihrer Arbeit geben Sie die KI als Quelle an, z.B.: Erzeugt durch Microsoft Copilot am dd.mm.yyyy. Oder: Entnommen aus einem Dialog mit Perplexity vom dd.mm.yyyy. Oder: Absatz 2.3 wurde durch ChatGPT sprachlich geglättet.

[25] Beispiele hierfür sind u.a. die folgenden Arbeitsschritte: Generierung von Ideen, Konzeption der Arbeit, Literatursuche, Literaturanalyse, Literaturverwaltung, Auswahl von Methoden, Datensammlung, Datenanalyse, Generierung von Programmcodes

Erklärung[26]

Ich versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema: *Lakehouse mit DuckDB – Konzept und Umsetzung* 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.

19.01.2025

(Ort, Datum) (Unterschrift)

[26] Studierende, die vor dem 01.10.2024 an der DHBW immatrikuliert waren, verwenden die Erklärung. Studierende, die ab dem 01.10.2024 an der DHBW immatrikuliert werden, verwenden die Ehrenwörtliche Erklärung (siehe Folgeseite). Diese Fußnote und die nicht benötigte Erklärung/Ehrenwörtliche Erklärung ist zu löschen.