# 

# Univerzita Jana Evangelisty Purkyně v Ústí nad Labem (UJEP) Přírodovědecká fakulta (PřF), Katedra informatiky (KI)

**OLAP a Data mining (KI/ODM)**

# 

ds

P

**Seminární práce**

**OLAP a ClickHouse**

Ondřej Švorc (F23209)

Aplikovaná informatika  
OLAP a Data mining (KI/ODM)

27. 4. 2025

**Obsah**

[Zkratky 3](#_Toc196689833)

[Volba DBMS s OLAP podporou 3](#_Toc196689834)

[Prerekvizity 3](#_Toc196689835)

[Datová sada 5](#_Toc196689836)

[Čištění datové sady 6](#_Toc196689837)

[Datová kostka 6](#_Toc196689838)

[Import datové sady 7](#_Toc196689839)

[Tvorba schématu hvězdy 8](#_Toc196689840)

[Komunikace s ClickHouse 9](#_Toc196689841)

[Řezy datovou kostkou 17](#_Toc196689842)

[Data mining 22](#_Toc196689843)

[Závěr 24](#_Toc196689844)

[Zdroje 24](#_Toc196689845)

## Zkratky

DBMS = Database Management System

OLAP = Online Analytical Processing

## Volba DBMS s OLAP podporou

Pro tento projekt jsem zvolil produkt **ClickHouse Cloud**, především díky jednoduchosti použití ve srovnání s ostatními DBMS s OLAP podporou. Před zvolením ClickHouse Cloud jsem si krátce prohlédl a vyzkoušel Azure Analytics Service, Apache Druid a SSMS.

## Prerekvizity

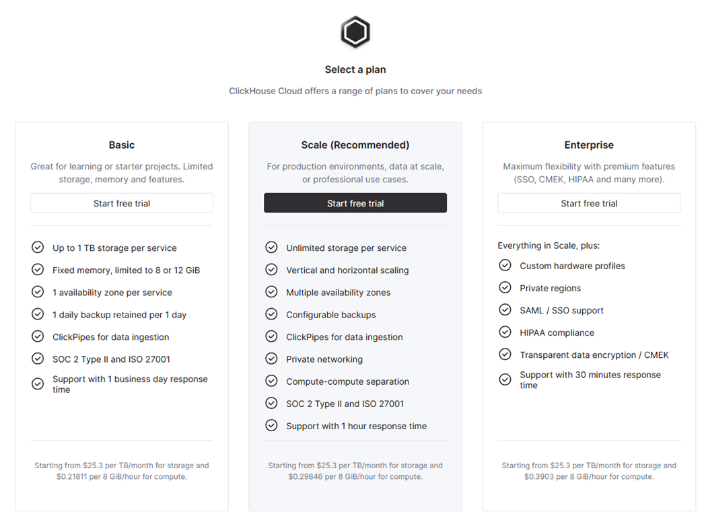
Díky tomu, že ClickHouse Cloud je webová aplikace, za jediné prerekvizity lze považovat:

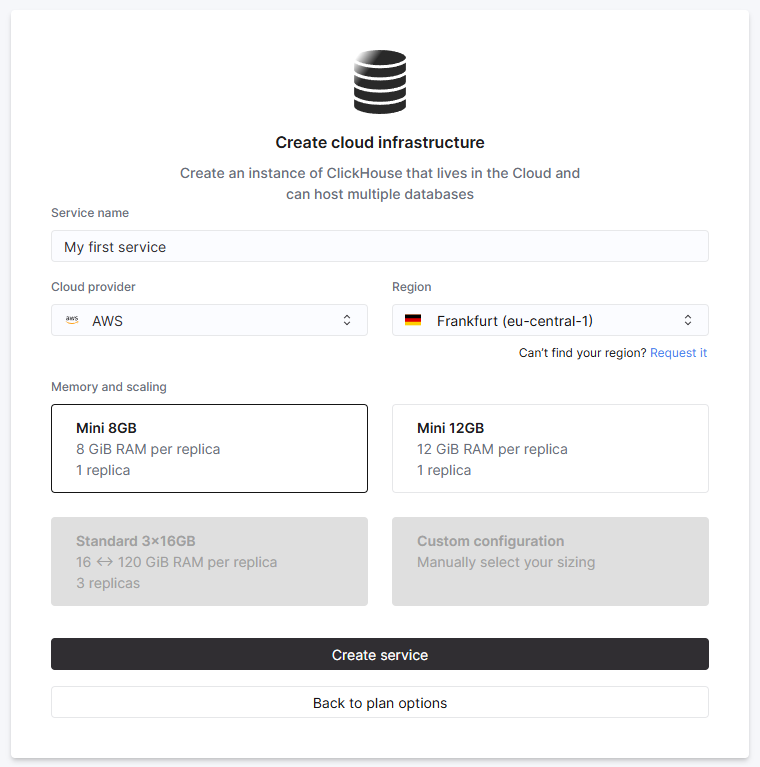
1. Založený účet
2. Zvolené předplatné
3. Vytvořená cloudová služba pro provoz

Osobně jsem zvolil možnost registrace pomocí **Google účtu** a předplatné **Basic**. Při vytváření cloudové služby jsem ponechal název **My first Service**, vybral poskytovatele **AWS**, lokaci ve **Frankfurtu nad Mohanem** (zatím nejbližší možná) a **8 GB RAM**.

Obsah obrázku text, snímek obrazovky, software, Operační systém

Obsah vygenerovaný umělou inteligencí může být nesprávný.





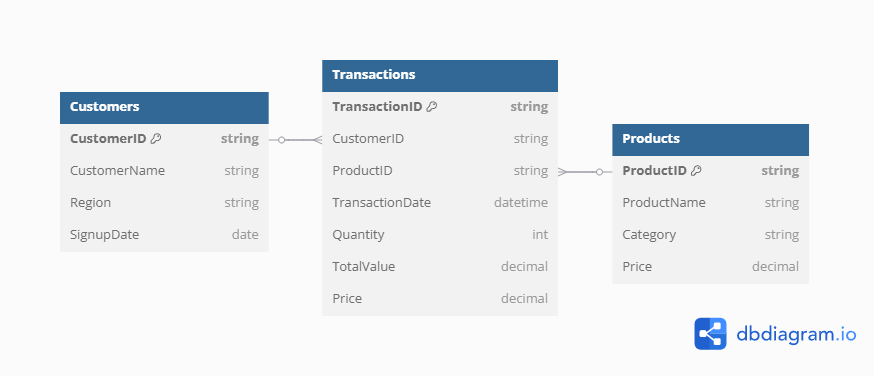
## Datová sada

Zvolil jsem datovou sadu ze stránky **Kaggle** fiktivního online obchodu **EverMart Online**.[[1]](#footnote-1)

Soubory datové sady ve formátu CSV obsahují data o zákaznících, produktech a provedených transakcí.

* Customers.csv
* Products.csv
* Transactions.csv

Provedené transakce jsou duplicitně zapsány ve dvou souborech s mírně odlišnými formáty (Transactions.csv, EverMart\_Online\_Transactions.xlsx). CSV formát je vhodnější pro strojové zpracování, a proto nemá smysl XLSX soubor zahrnovat.

Pro lepší představuje je zde ERD diagram, který reflektuje provázanost dat v CSV souborech.

## Čištění datové sady

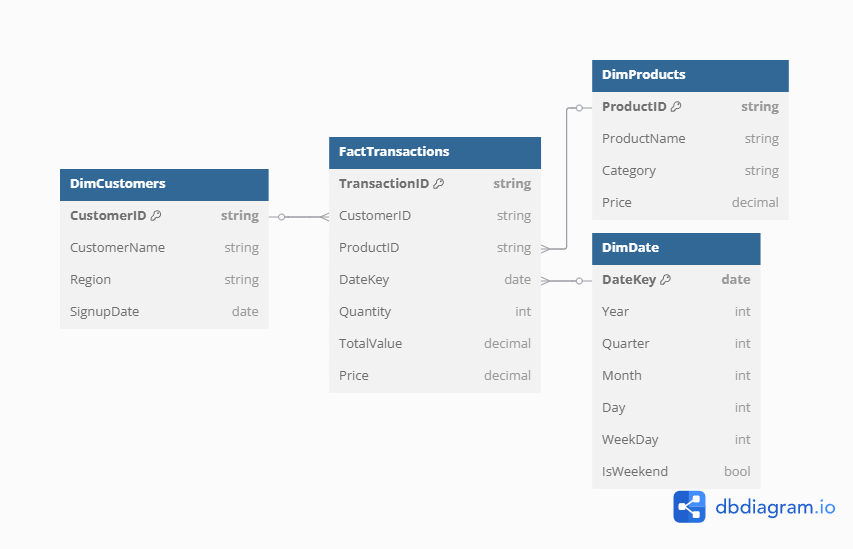
Datová sada byla již po stažení vyčištěná, a proto nebylo nutné provádět její čištění.

## Datová kostka

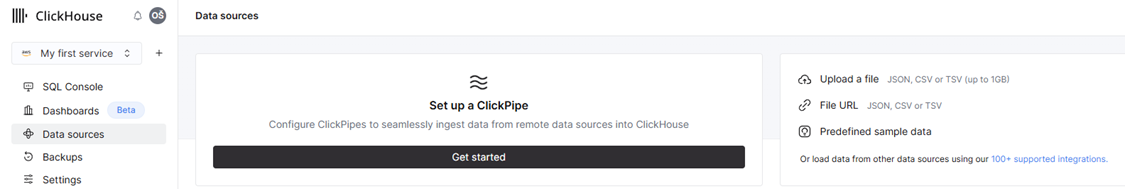
Zvolil jsem **schéma hvězdy (**star schema**)** se třemi dimenzemi a jednou faktovou tabulkou.

* DimCustomers – zákazníci (celé jméno, kontinent, datum registrace)
* DimProducts – produkty (název, kategorie, cena)
* DimDate – datum transakce (rok, měsíc, den, víkend)
* FactTransactions – transakce (množství, celková cena, odkazy na dimenze)

Zde jsou základní otázky, na jenž si můžeme na první pohled zodpovědět:

* Kdo nakupoval
* Co nakupoval
* Kdy nakoupil
* Kolik toho nakoupil
* Za kolik nakoupil

## Import datové sady

V Původně jsem chtěl vložit CSV soubory přímo pomocí grafického rozhraní ClickHouse Cloud, ale tato možnost opakovaně selhala. Rozhodl jsem se tedy data naimportovat pomocí Python skriptu.

Zvažoval jsem využití knihovny **clickhouse-connect[[2]](#footnote-2)** pro Python, která slouží jako rozhraní pro komunikaci s ClickHouse, ale nakonec jsem ji nepoužil, protože k mým účelům nebyla většina jejich funkcionalit vůbec potřeba; jednalo by se tak o zbytečnou závislost. Zvolil jsem tedy jednodušší řešení, a to použití přímých HTTP požadavků za pomocí knihovny **requests** Pythonu.

## Tvorba schématu hvězdy

1. Smazání všech existujících tabulek.
2. Vytvoření surových tabulek pro import dat na základě CSV souborů.
3. Import dat ze souborů CSV do surových tabulek.
4. Vytvoření dimenzionálních tabulek.
5. Naplnění dimenzionálních tabulek daty z tabulek surových.
6. Vytvoření faktové tabulky.
7. Naplnění faktové tabulky daty.
8. Smazání surových tabulek.

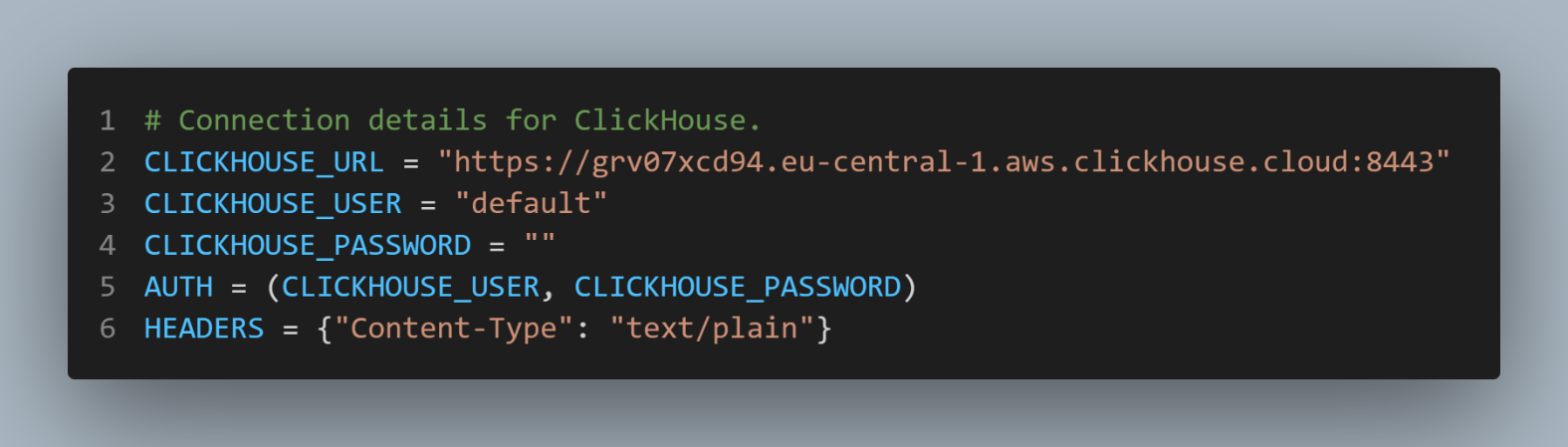
Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

## Komunikace s ClickHouse

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.Pro usnadnění opakovaného volání ClickHouse API jsem si na začátku skriptu definoval konstanty pro URL serveru, přihlašovací údaje a hlavičky HTTP požadavků.

Pro přehlednost jsem si také nadefinoval konstanty pro názvy surových tabulek, jejich přiřazení k CSV souborům, dimenzionální tabulky, faktovou tabulku a celkový seznam všech tabulek.

Obsah obrázku text, snímek obrazovky, displej, software

Obsah vygenerovaný umělou inteligencí může být nesprávný.Funkce **run\_sql** slouží k odesílání SQL příkazů na server ClickHouse, kde se SQL příkaz vyhodnotí a vykoná.  
Automaticky sestaví a odešle požadavek, zkontroluje jeho úspěšnost a vypíše stručnou informaci o výsledku.

Funkce **drop\_tables\_if\_exist** postupně smaže všechny tabulky uvedené v seznamu, pokud existují.

Tím zajišťuje, že při opětovném spuštění skriptu nedojde ke konfliktu kvůli existujícím tabulkám, a každý import proběhne vždy na čisté databázové struktuře.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Funkce **create\_raw\_tables** vytváří tři surové tabulky (Customers, Products, Transactions) podle struktury CSV souborů. Tyto tabulky slouží jako první krok před rozdělením dat do dimenzí a faktové tabulky.

Obsah obrázku text, snímek obrazovky, menu, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obsah obrázku text, snímek obrazovky, software, displej

Obsah vygenerovaný umělou inteligencí může být nesprávný.Funkce **insert\_csv\_data\_into\_raw\_tables** načítá obsah CSV souborů a vkládá jej do odpovídajících tabulek.

Funkce **create\_dim\_tables** vytvoří dimenzionální tabulky.

Obsah obrázku text, menu, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Funkce **insert\_into\_dim\_tables** vloží data ze surových tabulek do dimenzionálních.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obsah obrázku text, snímek obrazovky, Písmo, číslo

Obsah vygenerovaný umělou inteligencí může být nesprávný.Funkce **create\_fact\_table** vytvoří faktovou tabulku.

Funkce **insert\_into\_fact\_table** vloží a přetransformuje data z tabulky Transactions.

Obsah obrázku text, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

## Řezy datovou kostkou

Pro demonstraci práce s datovou kostkou jsem vytvořil následující čtyři řezy (a dva bonusové).

Obsah obrázku text, snímek obrazovky, vizitka, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obsah obrázku text, vizitka, snímek obrazovky, Písmo

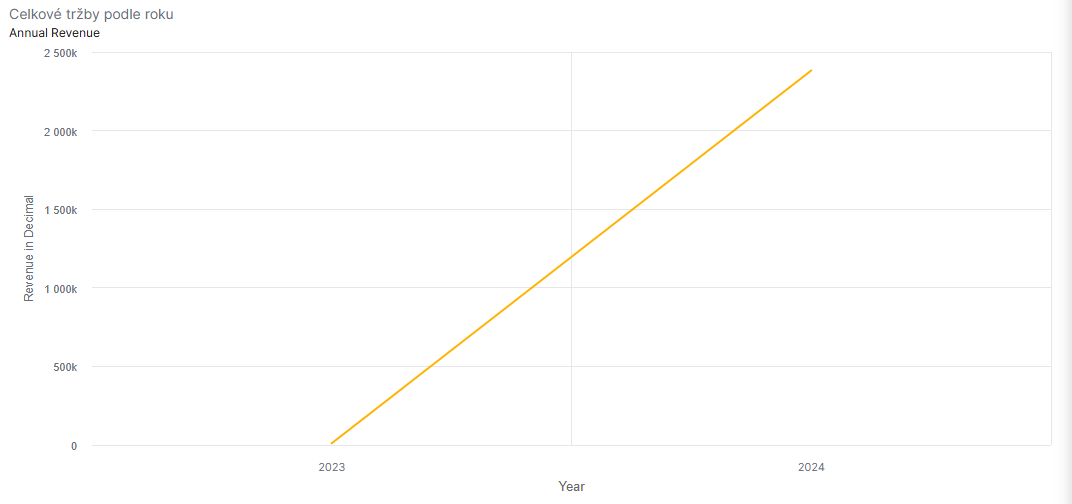
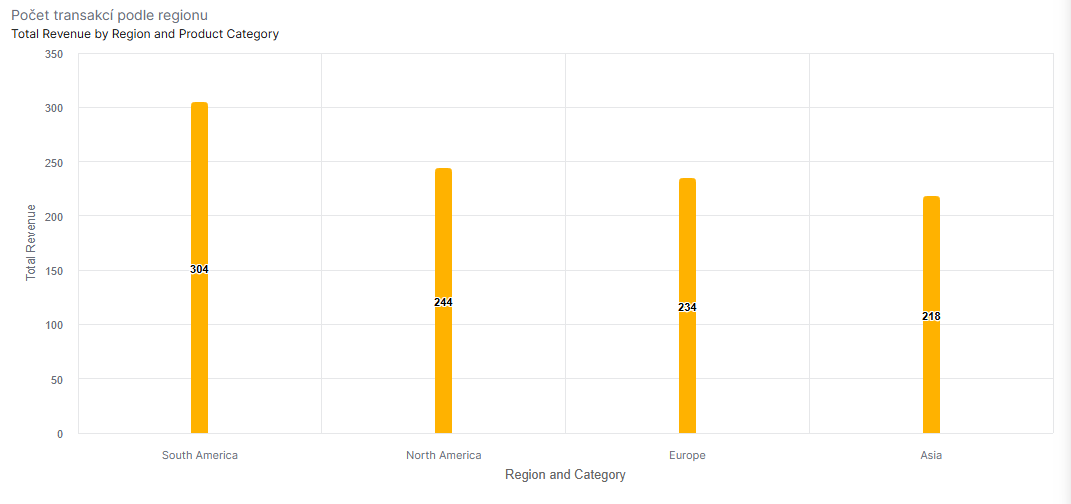
Obsah vygenerovaný umělou inteligencí může být nesprávný.

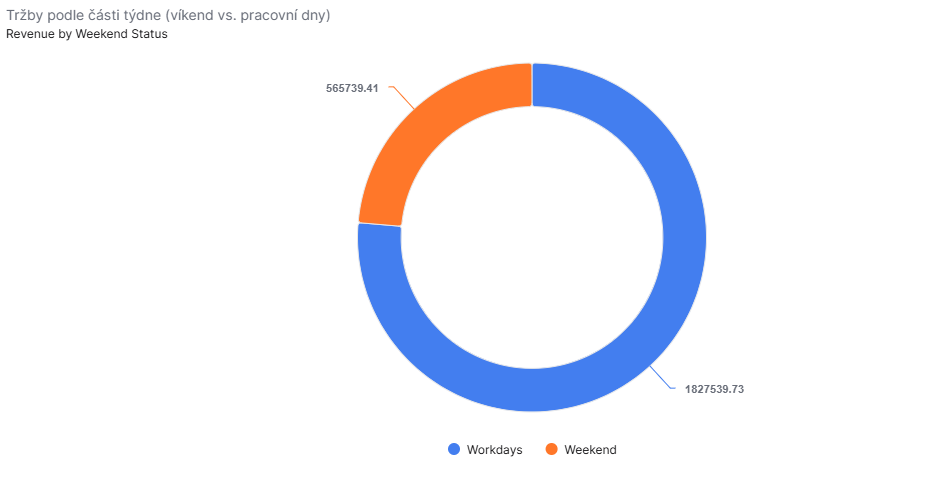
Obsah obrázku text, snímek obrazovky, Písmo, vizitka

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obsah obrázku text, vizitka, snímek obrazovky, Písmo

Obsah vygenerovaný umělou inteligencí může být nesprávný.



Obsah obrázku snímek obrazovky, diagram, text, Barevnost

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obsah obrázku text, snímek obrazovky, diagram, Obdélník

Obsah vygenerovaný umělou inteligencí může být nesprávný.Obsah obrázku text, snímek obrazovky, Písmo, vizitka

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Obsah obrázku text, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.Obsah obrázku text, Písmo, snímek obrazovky

Obsah vygenerovaný umělou inteligencí může být nesprávný.

## Data mining

ClickHouse nepodporuje nativní funkce pro data mining, a proto jsem se rozhodl tento krok realizovat pomocí Pythonu. V rámci úlohy jsem zvolil **shlukování (clustering)** metodou **K-means[[3]](#footnote-3)**, kdy jsem zákazníky rozdělil do tří skupin podle počtu nákupů a celkové útraty. Pro přípravu dat jsem v ClickHouse vygeneroval agregovaný dataset (CustomerID, PurchaseCount, TotalSpent), který jsem následně exportoval do souboru CSV.  
V Pythonu jsem tento soubor načetl a pomocí připraveného skriptu provedl shlukování a vizualizaci výsledků​.



Obsah obrázku text, snímek obrazovky, diagram, Vykreslený graf

Obsah vygenerovaný umělou inteligencí může být nesprávný.

Pomocí funkce print\_customers\_in\_group si lze vypsat, jací zákazníci se nachází v dané skupině, resp. jaký mají unikátní identifikátor.

## Závěr

Pomocí ClickHouse Cloud jsem navrhl a realizoval datovou kostku nad fiktivní e-commerce datovou sadou.  
Zvolil jsem schéma hvězdy, provedl import dat přes vlastní Python skript s využitím ClickHouse API a vytvořil několik analytických řezů kostkou. V části data mining jsem ukázal jednoduché shlukování zákazníků metodou K-means v Pythonu.

## Zdroje

**WAMBLES, Chad.** Ecommerce Transactions [online]. [cit. 25. 3. 2025]. Dostupné z: <https://www.kaggle.com/datasets/chadwambles/ecommerce-transactions>

**CLICKHOUSE, Inc.** ClickHouse Python Integrations [online]. [cit. 27. 4. 2025]. Dostupné z: <https://clickhouse.com/docs/integrations/python>

SCIKIT-LEARN Developers**.** KMeans clustering [online]. [cit. 27. 4. 2025]. Dostupné z: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

1. **WAMBLES, Chad.** Ecommerce Transactions [online]. [cit. 25. 3. 2025]. Dostupné z: <https://www.kaggle.com/datasets/chadwambles/ecommerce-transactions> [↑](#footnote-ref-1)
2. **CLICKHOUSE, Inc.** ClickHouse Python Integrations [online]. [cit. 27. 4. 2025]. Dostupné z: <https://clickhouse.com/docs/integrations/python> [↑](#footnote-ref-2)
3. SCIKIT-LEARN Developers**.** KMeans clustering [online]. [cit. 27. 4. 2025]. Dostupné z: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html> [↑](#footnote-ref-3)