

Univerzita Jana Evangelisty Purkyně v Ústí nad Labem

Přírodovědecká Fakulta

Katedra informatiky

Aplikovaná informatika

Zápočtový projekt z předmětu KI/ODM

**Analýza prodejních dat jízdních kol pomocí OLAP
modelu**

Student: David Král

Akademický rok: 2024/2025

Datum odevzdání: 5. 5. 2025

1. Úvod

Projekt řeší zpracování a analýzu rozsáhlé datové sady o prodeji jízdních kol. Cílem bylo vytvořit datový sklad, provést transformaci dat, sestavit hvězdicovou strukturu (fact-dimension model), realizovat analytické dotazy a nahradit data mining metodiku pokročilými SQL dotazy.

2. Použité nástroje

- Microsoft SQL Server Management Studio (SSMS)
- Microsoft SQL Server 2022
- CSV dataset: bike_sales.csv (přes 450 000 záznamů)
- Python pro import a předzpracování dat

3. Postup řešení

a) Import dat

CSV soubor byl naimportován do SQL Serveru přes Python (pandas + pyodbc). Data byla vyčištěna, konvertována a tabulka bike_sales_with_id byla doplněna o primární klíč (ID INT IDENTITY).

b) Transformace typů

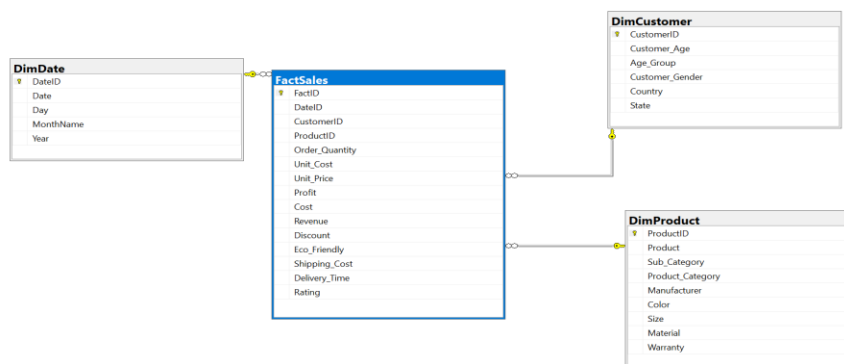
Pomocí ALTER COLUMN a UPDATE byly sloupce převedeny na odpovídající typy (DATE, INT, FLOAT, BIT).

c) Vytvoření dimenzí a faktové tabulky

Z tabulky bike_sales_with_id byla vytvořena:

- DimDate (datum, den, měsíc, rok)
- DimCustomer (věk, pohlaví, stát, skupina)
- DimProduct (produkt, kategorie, značka, velikost)
- FactSales (tržby, náklady, množství, odkaz na dimenze)

Faktová tabulka byla naplněna pomocí JOIN mezi zdrojovou tabulkou a dimenzemi.




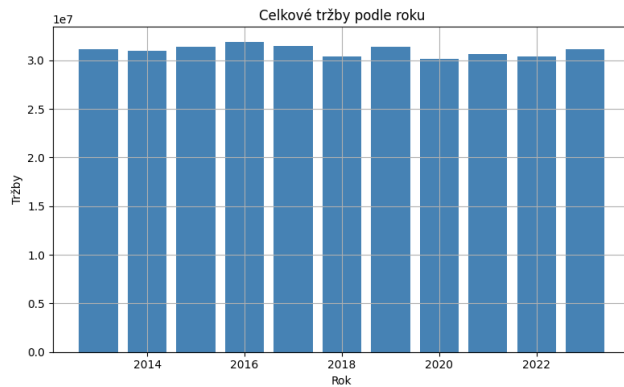
4. OLAP dotazy (bod 4b)

[OLAP/sql_dotazy_komentare.md](#) DavidKral9/OLAP

Dotaz 1: Tržby podle roku


```
SELECT
    d.Year,
    SUM(f.Revenue) AS TotalRevenue
FROM FactSales f
JOIN DimDate d ON f.DateID = d.DateID
GROUP BY d.Year
ORDER BY d.Year;
```

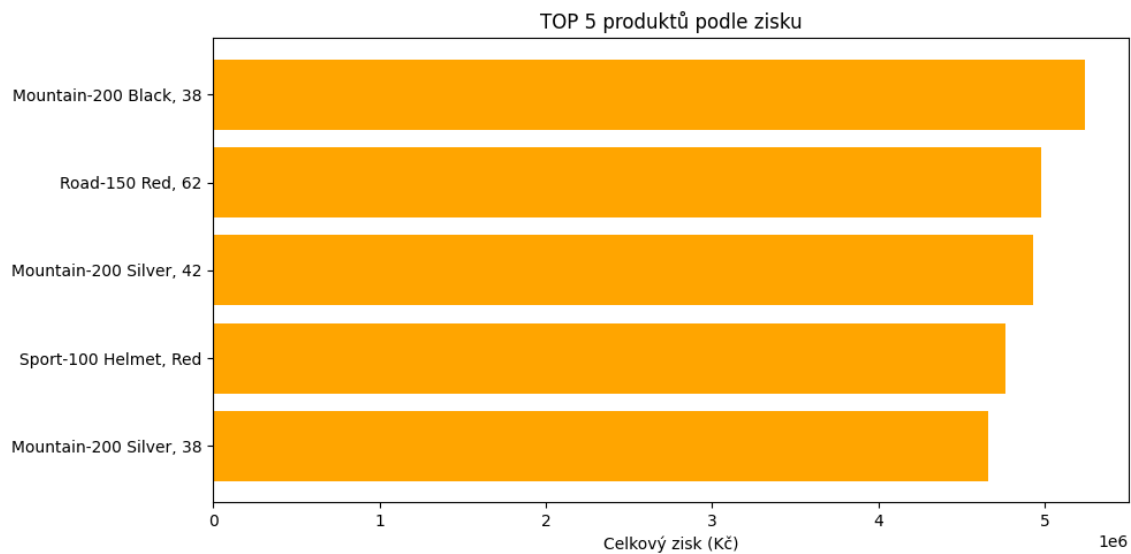
 Vrací součet tržeb (Revenue) seskupený podle kalendářního roku.



Dotaz 2: TOP 5 produktů podle zisku


```
SELECT
    p.Product,
    SUM(f.Profit) AS TotalProfit
FROM FactSales f
JOIN DimProduct p ON f.ProductID = p.ProductID
GROUP BY p.Product
ORDER BY TotalProfit DESC
OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;
```

 Vrací pět produktů s nejvyšším celkovým ziskem.

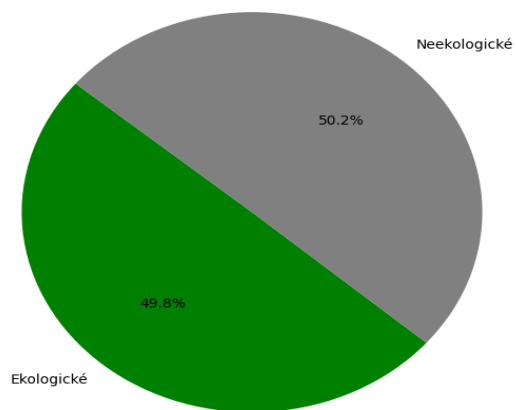


Dotaz 3: Podíl ekologických produktů

```
SELECT
    Eco_Friendly,
    COUNT(*) AS NumberOfOrders,
    SUM(Revenue) AS TotalRevenue
FROM FactSales
GROUP BY Eco_Friendly;
```


 Porovnává počet a součet tržeb objednávek mezi ekologickými a neekologickými produkty.

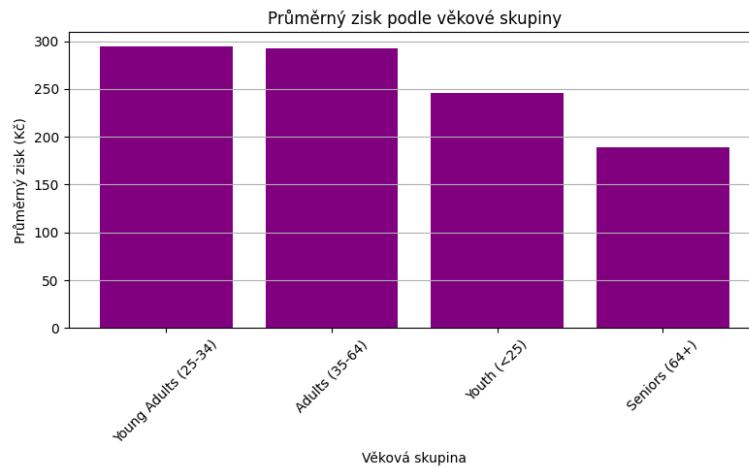
Podíl tržeb ekologických a neekologických produktů



Dotaz 4: Průměrný zisk podle věkových skupin

```
SELECT
    c.Age_Group,
    AVG(f.Profit) AS AverageProfit
FROM FactSales f
JOIN DimCustomer c ON f.CustomerID = c.CustomerID
GROUP BY c.Age_Group
ORDER BY AverageProfit DESC;
```

 Ukazuje, které věkové skupiny zákazníků generují nejvyšší průměrný zisk.



5. Pokročilý dotaz (náhrada data miningu – bod 4c)

A) Segmentace zákazníků podle průměrné hodnoty objednávky (High/Mid/Low):

Pomocí CTE dotazu byly zákazníci rozděleni do segmentů:


- High-Value (> 1000)
- Mid-Value (500–1000)
- Low-Value (< 500)

Dotaz využívá AVG, COUNT, CASE a GROUP BY.

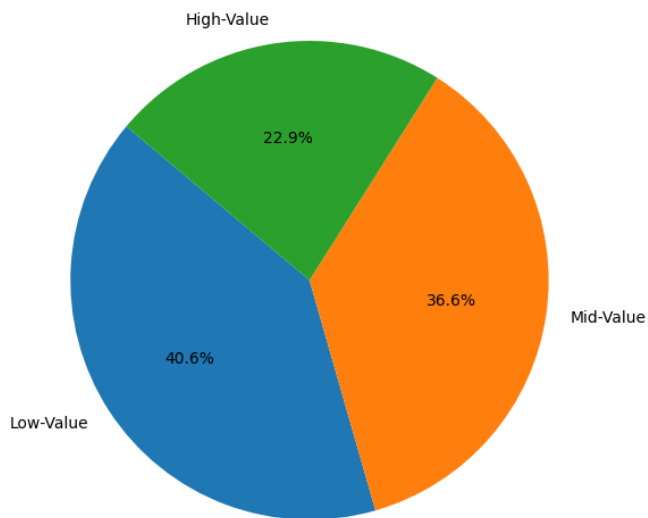
```

WITH CustomerAverages AS (
    SELECT
        c.CustomerID,
        c.Age_Group,
        c.Country,
        COUNT(f.FactID) AS OrderCount,
        SUM(f.Revenue) AS TotalRevenue,
        AVG(f.Revenue) AS AvgOrderValue
    FROM FactSales f
    JOIN DimCustomer c ON f.CustomerID = c.CustomerID
    GROUP BY c.CustomerID, c.Age_Group, c.Country
),
CustomerSegments AS (
    SELECT *,
        CASE
            WHEN AvgOrderValue > 1000 THEN 'High-Value'
            WHEN AvgOrderValue BETWEEN 500 AND 1000 THEN 'Mid-Value'
            ELSE 'Low-Value'
        END AS Segment
    FROM CustomerAverages
)
SELECT
    Segment,
    COUNT(*) AS NumberOfCustomers,
    AVG(AvgOrderValue) AS SegmentAvgOrder,
    SUM(TotalRevenue) AS SegmentRevenue
FROM CustomerSegments
GROUP BY Segment
ORDER BY SegmentAvgOrder DESC;

```

 Rozděluje zákazníky do tří segmentů podle hodnoty průměrné objednávky a zobrazuje souhrnné statistiky pro každý segment.

Segmentace zákazníků podle průměrné hodnoty objednávky



B) Decision tree

Co strom vyjadřuje

Kořen stromu: Revenue \leq 12.5

- první kritérium, které rozděluje všechny záznamy (452 144)
- pokud tržba z objednávky je \leq **12.5**, jdeme vlevo → většina neekologických
- pokud $>$ **12.5**, jdeme vpravo → možnost ekologických

Větev vlevo – nízké tržby → spíše neekologické

- Pokud je Order_Quantity ≤ 4.5 a Revenue ≤ 2.5 → **pravděpodobně neekologický**
- Pokud Rating ≤ 1.5 , spíše **neekologický**
- Obecně: **malé množství + nízký příjem = neekologické**

Větev vpravo – vyšší tržby → šance na ekologičnost

- Pokud Revenue ≤ 5999 , jdeme stále vlevo (např. hodně objednávek, ale levné produkty)
- Větve s Revenue > 6000 mají větší šanci být **ekologické**
- Například:
 - Revenue > 4006 → 280 216 ekologických vs. 208 104 neekologických
 - Revenue > 8701 → 720 ekologických vs. 668 neekologických

Atribut	Význam
gini	míra čistoty uzlu (čím blíží 0, tím lepší rozdělení)
samples	kolik záznamů je v daném uzlu
value = [0, 1]	kolik je neekologických / ekologických
class = ...	výsledná predikce uzlu
	○

Co z toho plyne:

- **Nízká cena a malé množství** → často **neekologické**
- **Větší tržby** → **vyšší pravděpodobnost ekologičnosti**
- Rating má určitý vliv, ale ne dominantní

```

import pyodbc
import pandas as pd
from sklearn.tree import DecisionTreeClassifier, plot_tree
import matplotlib.pyplot as plt

# Připojení k databázi
conn = pyodbc.connect(
    'DRIVER={ODBC Driver 17 for SQL Server};'
    'SERVER=LAPTOP-UIDJILKD;'
    'DATABASE=BikeSales;'
    'Trusted_Connection=yes;'
)

# Načteme data
query = """
SELECT Order_Quantity, Unit_Cost, Revenue, Rating, Eco_Friendly
FROM FactSales
"""
df = pd.read_sql(query, conn)

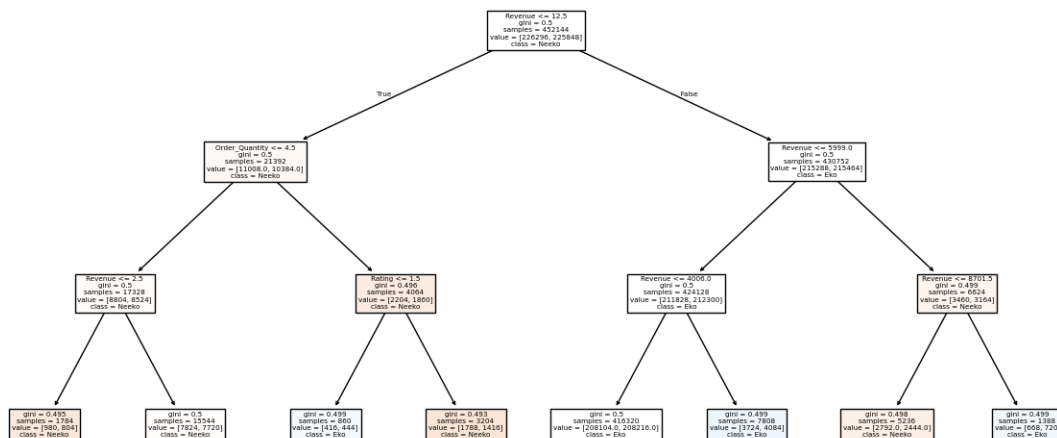
# Připravíme data
X = df.drop("Eco_Friendly", axis=1)
y = df["Eco_Friendly"]

# Model rozhodovacího stromu
model = DecisionTreeClassifier(max_depth=3)
model.fit(X, y)

# Vykreslení stromu
plt.figure(figsize=(12, 6))
plot_tree(model, feature_names=X.columns, class_names=["Neeko", "Eko"], filled=True)
plt.title("Rozhodovací strom: Ekologický vs. neekologický produkt")
plt.tight_layout()
plt.savefig("decision_tree.png")

```

Rozhodovací strom: Ekologický vs. neekologický produkt



6. Závěr

Projekt ukázal praktické použití relačního datového skladu a OLAP přístupů při analýze rozsáhlého datasetu. Byly vytvořeny dimenze, faktová tabulka, výkonné dotazy a pokročilé segmentace, které plně pokrývají požadavky zadání.

7. Zdroje

1. Microsoft Corporation. SQL Server Documentation. Dostupné z: <https://learn.microsoft.com/sql/>
2. Kimball, R. a Ross, M. The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling. 3rd Edition. Wiley, 2013. ISBN: 978-1118530801
3. Han, J., Pei, J., & Kamber, M. Data Mining: Concepts and Techniques. 3rd Edition. Morgan Kaufmann, 2011. ISBN: 978-0123814791
4. Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011. <https://scikit-learn.org/>
5. Seaborn: Statistical data visualization. Dostupné z: <https://seaborn.pydata.org/>
6. Python Software Foundation. Python Language Reference. Dostupné z: <https://www.python.org/doc/>
7. Visual Studio Documentation. Microsoft Docs. <https://learn.microsoft.com/en-us/visualstudio/>
8. Dodaný dataset Global Bike Sales Dataset (2013–2023).
9. Interní nápověda SQL Server Management Studio a oficiální dokumentace Microsoft Learn.