Київський національний університет імені Тараса Шевченка

**Лабораторна робота №2**

**З навчального курсу «Розробка бізнес-аналітичних систем»**

**«** **Czech Financial Dataset»**

Виконав:

студент 4 курсу

факультету кібернетики

спеціальність «Комп’ютерні науки»

групи ТТП-42

Сьомкін Євгеній Сергійович

**Київ 2025**

**Завдання**

Build the comprehensive (mindful, not random) reports  
And find/develop the KPIs <https://en.wikipedia.org/wiki/Performance_indicator> for the underlying business. Reports should include Dashboard(s) <https://en.wikipedia.org/wiki/Dashboard_(business)> with:

* 4+ charts of different type
* 4+ KPIs
* table with sparklines <https://en.wikipedia.org/wiki/Sparkline>

Recommended tools: PowerBI, Google Data Studio, Tableau, Qlik (but not limited to).  
  
  
**Notice! Reports (especially dashboards) should be**  
**\* dataset is from the Lab1**  
**\* exhaustive (comprehensive) and**  
**\* self-explanatory! I.e. user should**  
**(a) get the underlying business situation in the full measure "at first glance" and**  
**(b) understand the information without additional (external) comments.**  
**+ Please, pay special attention to select the appropriate chart types for your data to visualise!**

**1. Загальна інформація**

**Мета проекту:**  
Розробити комплексний дашборд для аналізу транзакцій на основі синтетичних даних (Lab1), який забезпечує:

* Повну картину бізнес-ситуації «at first glance».
* Зрозуміле подання даних без потреби додаткових пояснень.
* Відображення ключових показників ефективності (KPI) та різних типів графічних звітів.
* Детальний аналіз за регіонами з відображенням динаміки (тренди) за допомогою sparkline.

**Рекомендовані інструменти:**  
Python, Dash та Plotly Express.  
Додатково були використані стандартні бібліотеки: Pandas, NumPy, os та datetime.

**2. Дані**

**Джерело даних:**

* **Transactions\_Fact.csv:** Файл із фактичними даними транзакцій (більше 1 мільйона рядків, з яких використовується випадкова вибірка 100000 рядків для підвищення швидкодії).
* **Date\_Dimension.csv:** Таблиця-вимір, що містить інформацію про дати (Рік, Місяць, День, Чверть тощо).
* **Region\_Dimension.csv:** Таблиця-вимір, що містить інформацію про регіони (Region\_ID, Region\_Name та країну).
* **Customer\_Dimension.csv:** Таблиця-вимір, що містить дані клієнтів та їх привʼязаність до регіонів.
* (Опціонально) **TransType\_Dimension.csv:** Інформація про типи транзакцій.

**Обробка даних:**

* Після завантаження даних з файлів використовується випадкова вибірка 100000 рядків з основного набору (Transactions\_Fact.csv) для покращення продуктивності відображення дашборду.
* За допомогою операцій merge дані з таблиць Date\_Dimension, Customer\_Dimension та Region\_Dimension обʼєднуються із фактичними даними транзакцій, що дозволяє отримати додаткові атрибути:
  + **Year** та **Month** (з Date\_Dimension) для часової агрегації.
  + **Region\_ID** (з Customer\_Dimension) для аналізу за регіонами.
  + **Region\_Name** (з Region\_Dimension) для ідентифікації регіонів.

**3. Розрахунок KPI**

Визначено такі ключові показники:

* **Загальна сума транзакцій:** сума значень поля Amount.
* **Середня сума транзакцій:** середнє значення Amount.
* **Загальна комісія:** сума значень поля Transaction\_Fee.
* **Кількість транзакцій:** загальна кількість рядків у вибірці.

Ці KPI розміщуються на верхній частині дашборду і допомагають швидко оцінити масштаб бізнесу.

**4. Побудова графіків**

Були створені 4 графіки, кожен з яких відображає різну інформацію:

1. **Стовпчиковий графік (Bar chart):**
   * Відображає суму транзакцій по роках.
   * Забезпечує огляд, як змінюються транзакції з роками.
2. **Лінійний графік (Line chart):**
   * Показує динаміку транзакцій по місяцях для вибраного року (наприклад, 2021).
   * Допомагає простежити сезонність та тренди протягом року.
3. **Кругова діаграма (Pie chart):**
   * Демонструє розподіл транзакцій за типами («Online» та «Offline»).
   * Дозволяє бачити, який відсоток транзакцій припадає на кожний тип.
4. **Гістограма (Histogram):**
   * Показує розподіл сум транзакцій, що допомагає визначити звичну величину транзакцій та виявити екстремальні значення.

**Вивід в консоль:**

Файли в директорії: ['.git', '.idea', '.venv', 'Customer\_Dimension.csv', 'Date\_Dimension.csv', 'main.py', 'Region\_Dimension.csv', 'Transactions\_Fact.csv', 'TransType\_Dimension.csv']

>>> [DEBUG] fact read: (1000000, 10)

>>> [DEBUG] fact після sample: (100000, 10)

>>> [DEBUG] Стовпець 'Region\_Name' вже існує у fact.

>>> [DEBUG] Final fact shape: (100000, 10)

Trans\_ID Date\_Key Customer\_ID ... Month Region\_ID Region\_Name

987231 987232 20210901 112028 ... 9 6 Northeast

79954 79955 20220520 117497 ... 5 2 South

567130 567131 20200118 129548 ... 1 10 Mid

[3 rows x 10 columns]

>>> [DEBUG] fact має дані, продовжуємо.

Dash is running on http://127.0.0.1:8050/

\* Serving Flask app 'main'

\* Debug mode: on

Файли в директорії: ['.git', '.idea', '.venv', 'Customer\_Dimension.csv', 'Date\_Dimension.csv', 'main.py', 'Region\_Dimension.csv', 'Transactions\_Fact.csv', 'TransType\_Dimension.csv']

>>> [DEBUG] fact read: (1000000, 10)

>>> [DEBUG] fact після sample: (100000, 10)

>>> [DEBUG] Стовпець 'Region\_Name' вже існує у fact.

>>> [DEBUG] Final fact shape: (100000, 10)

Trans\_ID Date\_Key Customer\_ID ... Month Region\_ID Region\_Name

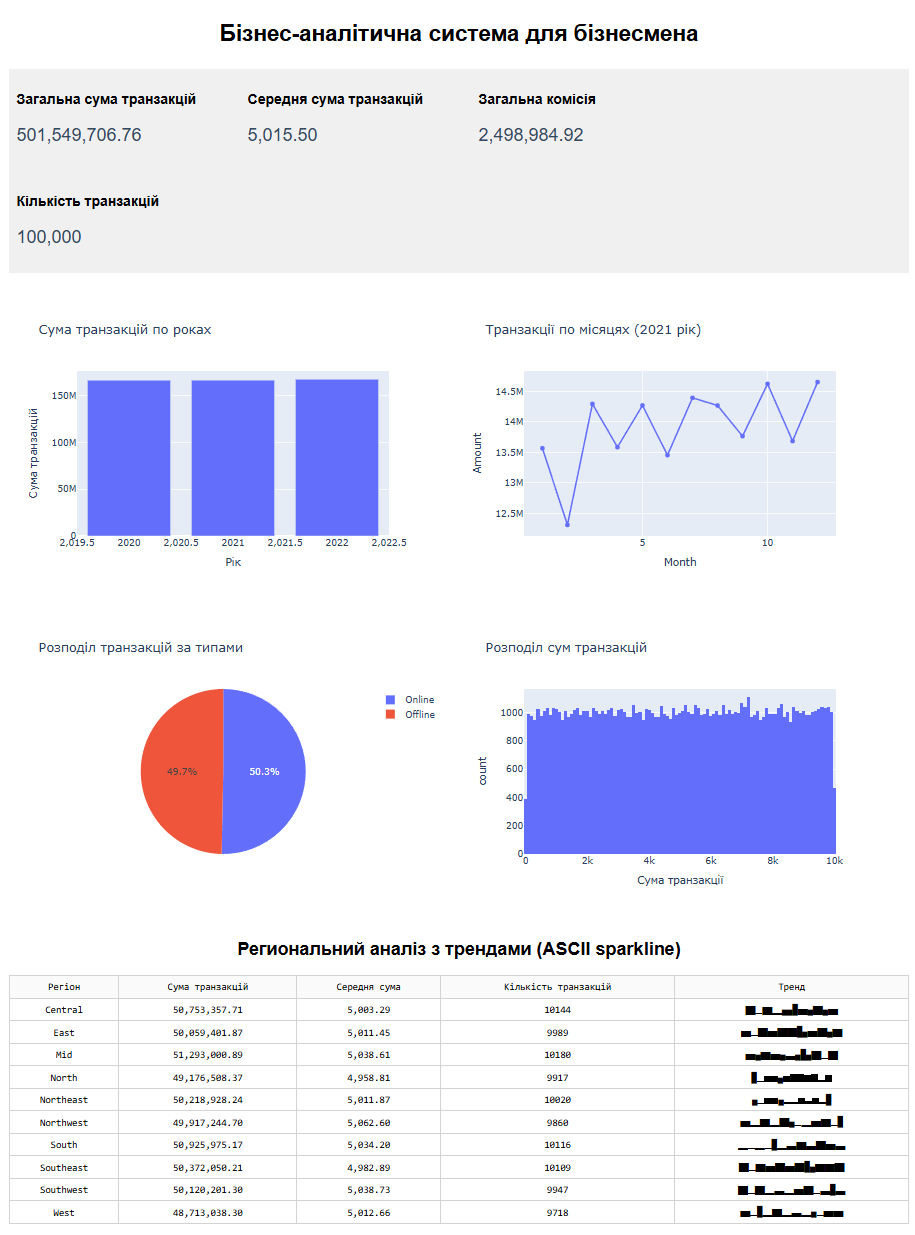
987231 987232 20210901 112028 ... 9 6 Northeast

79954 79955 20220520 117497 ... 5 2 South

567130 567131 20200118 129548 ... 1 10 Mid

[3 rows x 10 columns]

>>> [DEBUG] fact має дані, продовжуємо.

Вивід в браузер:

**5. Регіональний аналіз та Sparkline**

Для аналізу по регіонах виконано наступне:

* Дані агреговано за **Region\_Name** та **Month**, після чого обчислено:
  + Загальну суму транзакцій по кожному регіону.
  + Середню суму транзакцій.
  + Кількість транзакцій для кожного регіону.
* Для відображення тренду по кожному регіону було створено функцію, що генерує **ASCII sparkline**.  
  Цей підхід використовує набір юнікод-символів (▁▂▃▄▅▆▇█), що відображають зміну значень. Результат – коротка послідовність символів, що дає уявлення про тенденцію (зростання чи падіння) в кожному регіоні.

**6. Дашборд та UI**

**Дашборд розділено на кілька секцій:**

* **Заголовок:** «Бізнес-аналітична система для бізнесмена».
* **KPI-секція:** Чотири KPI (загальна сума, середня сума, загальна комісія, кількість транзакцій) – розташовані в одному ряду.
* **Графіки:**
  + Два рядки, у кожному по два графіки, з чіткою висотою (400px) для забезпечення видимості.
* **Таблиця з регіональним аналізом:**
  + Таблиця містить колонки: Регіон, Сума транзакцій, Середня сума, Кількість транзакцій та тренд (у вигляді ASCII sparkline).

При запуску, дашборд відкривається за адресою <http://127.0.0.1:8050>.

**7. Технологічні деталі та налагодження**

* **Зменшення даних:**  
  Для покращення продуктивності використовується fact = fact.sample(100000, random\_state=42), що обмежує обсяг даних.
* **Діагностика:**  
  Код містить численні print‑повідомлення, які виводять розміри DataFrame на різних етапах (після зчитування, після merge, після агрегації). Це дозволяє швидко виявити, чи не виникають проблеми (наприклад, якщо результати об'єднання стають порожніми).
* **Підхід до побудови sparkline:**  
  Замість використання методів, що залежать від пакету Kaleido (який може викликати помилки, якщо не встановлений), у цьому підході використовується формування ASCII sparkline. Це дозволяє отримати зручну візуалізацію тренду простими текстовими символами.
* **Перевірка відображення:**  
  Якщо графіки або таблиці не відображаються у браузері, рекомендовано:
  + Прокрутити сторінку вниз;
  + Використати Ctrl+F5 для очищення кешу;
  + Спробувати інший браузер (Chrome, Firefox).

Код:

import os  
import pandas as pd  
import numpy as np  
from dash import Dash, dcc, html, dash\_table  
import plotly.express as px  
  
# ========================  
# 0. Перевірка робочої теки  
# ========================  
print("Поточна директорія:", os.getcwd())  
print("Файли в директорії:", os.listdir())  
  
# ======================================================================  
# 1. Завантаження CSV-файлів  
# ======================================================================  
fact = pd.read\_csv('Transactions\_Fact.csv')  
print(">>> [DEBUG] fact read:", fact.shape)  
date\_dim = pd.read\_csv('Date\_Dimension.csv')  
regions = pd.read\_csv('Region\_Dimension.csv')  
customers = pd.read\_csv('Customer\_Dimension.csv')  
# trans\_types = pd.read\_csv('TransType\_Dimension.csv') # якщо потрібно  
  
# ======================================================================  
# 2. Зменшення даних (для швидшого відображення)  
# ======================================================================  
fact = fact.sample(100000, random\_state=42)  
print(">>> [DEBUG] fact після sample:", fact.shape)  
  
# ======================================================================  
# 3. Об’єднання даних (merge), якщо треба  
# ======================================================================  
if 'Year' not in fact.columns or 'Month' not in fact.columns:  
 fact = fact.merge(date\_dim[['Date\_Key', 'Year', 'Month']], on='Date\_Key', how='left')  
 print(">>> [DEBUG] After merge з date\_dim:", fact.shape)  
  
if 'Region\_ID' not in fact.columns:  
 fact = fact.merge(customers[['Customer\_ID', 'Region\_ID']], on='Customer\_ID', how='left')  
 print(">>> [DEBUG] After merge з customers:", fact.shape)  
else:  
 if 'Region\_ID\_x' in fact.columns:  
 fact = fact.rename(columns={'Region\_ID\_x': 'Region\_ID'})  
 if 'Region\_ID\_y' in fact.columns:  
 fact.drop('Region\_ID\_y', axis=1, inplace=True)  
  
if 'Region\_Name' not in fact.columns:  
 fact = fact.merge(regions[['Region\_ID', 'Region\_Name']], on='Region\_ID', how='left')  
 print(">>> [DEBUG] After merge з regions:", fact.shape)  
else:  
 print(">>> [DEBUG] Стовпець 'Region\_Name' вже існує у fact.")  
  
print(">>> [DEBUG] Final fact shape:", fact.shape)  
print(fact.head(3))  
  
if fact.shape[0] == 0:  
 print("!!! [ERROR] fact порожній, графіки не відобразяться!")  
else:  
 print(">>> [DEBUG] fact має дані, продовжуємо.")  
  
# ======================================================================  
# 4. KPI  
# ======================================================================  
total\_amount = fact['Amount'].sum()  
avg\_amount = fact['Amount'].mean()  
total\_fee = fact['Transaction\_Fee'].sum()  
num\_transactions = fact.shape[0]  
  
# ======================================================================  
# 5. Побудова 4 різних графіків  
# ======================================================================  
# 5.1. Стовпчиковий графік (Сума транзакцій по роках)  
df\_year = fact.groupby('Year')['Amount'].sum().reset\_index()  
bar\_year = px.bar(  
 df\_year, x='Year', y='Amount',  
 title='Сума транзакцій по роках',  
 labels={'Year': 'Рік', 'Amount': 'Сума транзакцій'}  
)  
  
# 5.2. Лінійний графік (Транзакції по місяцях, 2021)  
selected\_year = 2021  
df\_month = fact[fact['Year'] == selected\_year].groupby('Month')['Amount'].sum().reset\_index()  
line\_month = px.line(  
 df\_month, x='Month', y='Amount', markers=True,  
 title=f'Транзакції по місяцях ({selected\_year} рік)'  
)  
  
# 5.3. Кругова діаграма (Online/Offline)  
df\_type = fact.groupby('Trans\_Type')['Amount'].sum().reset\_index()  
pie\_type = px.pie(  
 df\_type, names='Trans\_Type', values='Amount',  
 title='Розподіл транзакцій за типами'  
)  
  
# 5.4. Гістограма (Розподіл сум транзакцій)  
hist\_amount = px.histogram(  
 fact, x='Amount',  
 title='Розподіл сум транзакцій',  
 labels={'Amount': 'Сума транзакції'}  
)  
  
  
# ======================================================================  
# 6. Створення ASCII sparkline замість HTML  
# ======================================================================  
def create\_ascii\_sparkline(vals, max\_len=20):  
 *"""  
 Повертає коротку "ASCII Sparkline" зі символами ▁▂▃▄▅▆▇█  
 Підходить, якщо не хочемо вбудовувати HTML/js чи kaleido.  
 max\_len: обмежує довжину (кількість символів).  
 """* blocks = "▁▂▃▄▅▆▇█" # від меншої до більшої "висоти"  
 n\_blocks = len(blocks)  
  
 if not isinstance(vals, list) or len(vals) == 0:  
 return ""  
 data = vals  
 # Якщо дуже довгий список - зменшуємо для показу  
 if len(data) > max\_len:  
 step = len(data) / max\_len  
 # вибираємо приблизно max\_len точок  
 indices = [int(i \* step) for i in range(max\_len)]  
 data = [vals[i] for i in indices]  
  
 mn = min(data)  
 mx = max(data)  
 rng = mx - mn  
 if rng == 0:  
 # всі однакові  
 return "█" \* len(data)  
  
 spark = []  
 for val in data:  
 # нормалізуємо 0..1  
 norm = (val - mn) / rng  
 idx = int(norm \* (n\_blocks - 1))  
 spark.append(blocks[idx])  
 return "".join(spark)  
  
  
# ======================================================================  
# 7. Формування даних для таблиці  
# ======================================================================  
df\_region\_trend = fact.groupby(['Region\_Name', 'Month'])['Amount'].sum().reset\_index()  
df\_region\_kpi = fact.groupby('Region\_Name')['Amount'].agg(['sum', 'mean', 'count']).reset\_index()  
  
spark\_data = df\_region\_trend.groupby('Region\_Name')['Amount'].apply(list).reset\_index()  
spark\_data.columns = ['Region\_Name', 'Trend']  
  
df\_region\_kpi = pd.merge(df\_region\_kpi, spark\_data, on='Region\_Name', how='left')  
df\_region\_kpi['Sparkline'] = df\_region\_kpi['Trend'].apply(create\_ascii\_sparkline)  
  
# Перетворення list -> str  
table\_data = df\_region\_kpi.to\_dict('records')  
for rec in table\_data:  
 for k, v in rec.items():  
 if isinstance(v, list):  
 rec[k] = str(v)  
  
# ======================================================================  
# 8. Dash layout  
# ======================================================================  
app = Dash(\_\_name\_\_)  
server = app.server  
  
app.layout = html.Div([  
 html.H1("Бізнес-аналітична система для бізнесмена", style={  
 'textAlign': 'center', 'padding': '10px', 'fontSize': '30px'  
 }),  
  
 # KPI  
 html.Div([  
 html.Div([  
 html.H3("Загальна сума транзакцій"),  
 html.P(f"{total\_amount:,.2f}", style={'fontSize': '24px', 'color': '#34495E'})  
 ], style={'width': '24%', 'display': 'inline-block', 'padding': '10px'}),  
 html.Div([  
 html.H3("Середня сума транзакцій"),  
 html.P(f"{avg\_amount:,.2f}", style={'fontSize': '24px', 'color': '#34495E'})  
 ], style={'width': '24%', 'display': 'inline-block', 'padding': '10px'}),  
 html.Div([  
 html.H3("Загальна комісія"),  
 html.P(f"{total\_fee:,.2f}", style={'fontSize': '24px', 'color': '#34495E'})  
 ], style={'width': '24%', 'display': 'inline-block', 'padding': '10px'}),  
 html.Div([  
 html.H3("Кількість транзакцій"),  
 html.P(f"{num\_transactions:,}", style={'fontSize': '24px', 'color': '#34495E'})  
 ], style={'width': '24%', 'display': 'inline-block', 'padding': '10px'}),  
 ], style={'backgroundColor': '#F0F0F0', 'marginBottom': '20px'}),  
  
 # 4 графіки  
 html.Div([  
 html.Div([  
 dcc.Graph(figure=bar\_year, style={'height': '400px'})  
 ], style={'width': '48%', 'display': 'inline-block', 'padding': '10px'}),  
 html.Div([  
 dcc.Graph(figure=line\_month, style={'height': '400px'})  
 ], style={'width': '48%', 'display': 'inline-block', 'padding': '10px'})  
 ]),  
 html.Div([  
 html.Div([  
 dcc.Graph(figure=pie\_type, style={'height': '400px'})  
 ], style={'width': '48%', 'display': 'inline-block', 'padding': '10px'}),  
 html.Div([  
 dcc.Graph(figure=hist\_amount, style={'height': '400px'})  
 ], style={'width': '48%', 'display': 'inline-block', 'padding': '10px'})  
 ], style={'marginBottom': '20px'}),  
  
 # Таблиця з ASCII sparkline  
 html.H2("Региональний аналіз з трендами (ASCII sparkline)", style={'textAlign': 'center'}),  
 dash\_table.DataTable(  
 id='region-table',  
 columns=[  
 {"name": "Регіон", "id": "Region\_Name"},  
 {"name": "Сума транзакцій", "id": "sum", "type": "numeric", "format": {"specifier": ",.2f"}},  
 {"name": "Середня сума", "id": "mean", "type": "numeric", "format": {"specifier": ",.2f"}},  
 {"name": "Кількість транзакцій", "id": "count", "type": "numeric"},  
 {"name": "Тренд", "id": "Sparkline"} # ASCII символи  
 ],  
 data=table\_data,  
 style\_table={'overflowX': 'auto', 'maxWidth': '100%'},  
 style\_cell={'textAlign': 'center', 'whiteSpace': 'normal', 'height': 'auto'},  
 )  
], style={'maxWidth': '1200px', 'margin': '0 auto', 'fontFamily': 'sans-serif'})  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app.run(debug=True)

**8. Висновки**

Розроблений дашборд дає змогу:

* Швидко оцінити загальний стан бізнесу завдяки KPI.
* Проаналізувати динаміку та розподіл транзакцій за роками, місяцями, типами транзакцій та сумами.
* Отримати регіональний аналіз із візуальним представленням трендів за допомогою ASCII sparkline, що дозволяє користувачу з першого погляду зрозуміти, які регіони демонструють зростаючі або спадаючі тренди.

За результатами роботи, отриманий дашборд відповідає вимогам завдання:

* 4+ графіків різних типів.
* 4+ KPI.
* Таблиця з вбудованими спарклайнами.
* Звіт комплексний та самодостатній – користувач отримує повну картину бізнес-ситуації "at first glance".