Київський національний університет імені Тараса Шевченка

**Лабораторна робота №1**

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

**«** **Czech Financial Dataset»**

Виконав:

студент 4 курсу

факультету кібернетики

спеціальність «Комп’ютерні науки»

групи ТТП-42

Чебан Богдан Володимирович

**Київ 2025**

## **Task (Завдання)**

**English:**

Develop a comprehensive Business Intelligence (BI) solution to analyze the 1999 Czech Bank Financial Dataset, which includes bank transactions, loans, and credit cards data. The solution should encompass data extraction, transformation, loading (ETL), data modeling using a star schema for an OLAP warehouse, and data visualization through interactive dashboards.

**Українська:**

Розробити комплексне рішення для бізнес-аналітики (BI) для аналізу фінансового набору даних чеського банку за 1999 рік, який включає дані про банківські транзакції, кредити та кредитні картки. Рішення повинно охоплювати витяг даних, їх трансформацію, завантаження (ETL), моделювання даних за допомогою зіркової схеми для OLAP-сховища та візуалізацію даних через інтерактивні дашборди.

## **Domain Short Description (Короткий опис домену)**

**English:**

The project focuses on financial data analysis within the banking sector. It aims to provide insights into customer transactions, loan distributions, and credit card usage patterns from the 1999 dataset. By leveraging BI tools, the solution facilitates informed decision-making and strategic planning for the bank.

**Українська:**

Проект зосереджений на аналізі фінансових даних у банківському секторі. Його мета — надати інсайти щодо транзакцій клієнтів, розподілу кредитів та моделей використання кредитних карток з набору даних за 1999 рік. Використовуючи інструменти BI, рішення сприяє прийняттю обґрунтованих рішень та стратегічному плануванню банку.

## **Original Dataset Link + Short Description (Оригінальне посилання на набір даних + короткий опис)**

### **English:**

* **Dataset:** 1999 Czech Bank Financial Dataset
* **Link:** <https://www.kaggle.com/datasets/marceloventura/the-berka-dataset/data>
* **Description:**  
  The **1999 Czech Bank Financial Dataset**, also known as **The Berka Dataset**, is a comprehensive collection of financial information from a Czech bank. Compiled by Petr Berka and Marta Sochorova, the dataset encompasses data related to over 5,300 bank clients and approximately 1,000,000 transactions conducted in the year 1999. Additionally, the dataset includes information on nearly 700 loans and about 900 credit cards issued to the clients.

**Key Components:**

**Transaction Data (trans.csv):**

* **Fields:**
* trans\_id: Transaction Identifier (Primary Key)
* date: Date of Transaction (Format: YYMMDD)
* account\_id: Account Identifier (Foreign Key to dim\_account)
* amount: Transaction Amount
* balance: Account Balance After Transaction
* client\_id: Client Identifier (Foreign Key to dim\_client)
* district\_id: District Identifier (Foreign Key to dim\_district)
* type\_of\_trans: Type of Transaction
* operation: Operation Details
* Additional fields related to loans and credit cards.

**Client Data (client.csv):**

* **Fields:**
* client\_id: Client Identifier (Primary Key)
* birth\_number: Client's Birth Number
* district\_id: District Identifier (Foreign Key to dim\_district)
* gender: Client's Gender
* birth\_date: Client's Birth Date
* Additional demographic and personal attributes.

**Account Data (account.csv):**

* **Fields:**
* account\_id: Account Identifier (Primary Key)
* frequency: Transaction Frequency
* date\_created: Account Creation Date
* Additional account-related attributes.

**District Data (district.csv):**

* **Fields:**
* district\_id: District Identifier (Primary Key)
* region: Geographic Region
* district\_name: Name of the District
* Additional geographic and administrative attributes.

**Order Data (order.csv):**

* **Fields:**
* order\_id: Order Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* order\_date\_id: Order Date Identifier (Foreign Key to dim\_date)
* order\_amount: Amount of the Order
* Additional order-related measures.

**Dispensation Data (disp.csv):**

* **Fields:**
* disp\_id: Dispensation Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* disp\_type: Type of Dispensation
* disp\_date\_id: Dispensation Date Identifier (Foreign Key to dim\_date)
* Additional dispensation-related measures.

**Loan Data (loan.csv):**

* **Fields:**
* loan\_id: Loan Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* loan\_date\_id: Loan Date Identifier (Foreign Key to dim\_date)
* loan\_amount: Amount of the Loan
* loan\_type: Type of Loan
* Additional loan-related measures.

**Credit Card Data (card.csv):**

* **Fields:**
* card\_id: Credit Card Identifier (Primary Key)
* client\_id: Client Identifier (Foreign Key to dim\_client)
* card\_type: Type of Credit Card
* card\_issue\_date: Issue Date of the Credit Card
* card\_limit: Credit Limit of the Card
* Additional credit card-related measures.

**Additional Information:**

* + **Data Volume:** The dataset includes over 1,000,000 transaction records, providing a robust foundation for extensive financial analysis.
  + **Dimensions and Measures:** The data is structured to support a multi-dimensional analysis with four primary Dimension Tables (dim\_date, dim\_client, dim\_district, dim\_account) and multiple Fact Tables (fact\_trans, fact\_order, fact\_disp, fact\_loan, fact\_card) capturing various financial measures.
  + **Hierarchies:** The dim\_date table incorporates hierarchies such as year, quarter, month, and day, facilitating time-based analyses. Similarly, geographic hierarchies are present in the dim\_district table.
  + **Usage:** This dataset is ideal for constructing an OLAP warehouse using a star schema, enabling efficient querying, reporting, and visualization of financial data for strategic decision-making.

### **Українська:**

* **Набір Даних:** Фінансовий набір даних чеського банку за 1999 рік
* **Посилання:** <https://www.kaggle.com/datasets/marceloventura/the-berka-dataset/data>
* **Опис:**  
  **Фінансовий набір даних чеського банку за 1999 рік**, також відомий як **The Berka Dataset**, є всебічним збором фінансової інформації від чеського банку. Складаний Петром Беркою та Мартією Сохровою, набір даних охоплює інформацію про понад 5,300 клієнтів банку та близько 1,000,000 транзакцій, проведених у 1999 році. Крім того, набір даних включає інформацію про майже 700 кредитів та близько 900 кредитних карток, виданих клієнтам.

**Ключові компоненти:**

**Транзакційні Дані (trans.csv):**

* **Поля:**
* trans\_id: Ідентифікатор транзакції (Первинний ключ)
* date: Дата транзакції (Формат: YYMMDD)
* account\_id: Ідентифікатор рахунку (Зовнішній ключ до dim\_account)
* amount: Сума транзакції
* balance: Баланс рахунку після транзакції
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* district\_id: Ідентифікатор району (Зовнішній ключ до dim\_district)
* type\_of\_trans: Тип транзакції
* operation: Деталі операції
* Додаткові поля, пов'язані з кредитами та кредитними картками.

**Дані про Клієнтів (client.csv):**

* **Поля:**
* client\_id: Ідентифікатор клієнта (Первинний ключ)
* birth\_number: Номер народження клієнта
* district\_id: Ідентифікатор району (Зовнішній ключ до dim\_district)
* gender: Стать клієнта
* birth\_date: Дата народження клієнта
* Додаткові демографічні та персональні атрибути.

**Дані про Рахунки (account.csv):**

* **Поля:**
* account\_id: Ідентифікатор рахунку (Первинний ключ)
* frequency: Частота транзакцій
* date\_created: Дата створення рахунку
* Додаткові атрибути, пов'язані з рахунками.

**Дані про Райони (district.csv):**

* **Поля:**
* district\_id: Ідентифікатор району (Первинний ключ)
* region: Географічний регіон
* district\_name: Назва району
* Додаткові географічні та адміністративні атрибути.

**Дані про Замовлення (order.csv):**

* **Поля:**
* order\_id: Ідентифікатор замовлення (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* order\_date\_id: Ідентифікатор дати замовлення (Зовнішній ключ до dim\_date)
* order\_amount: Сума замовлення
* Додаткові міри, пов'язані з замовленнями.

**Дані про Диспетчеризацію (disp.csv):**

* **Поля:**
* disp\_id: Ідентифікатор диспетчеризації (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* disp\_type: Тип диспетчеризації
* disp\_date\_id: Ідентифікатор дати диспетчеризації (Зовнішній ключ до dim\_date)
* Додаткові міри, пов'язані з диспетчеризацією.

**Дані про Кредити (loan.csv):**

* **Поля:**
* loan\_id: Ідентифікатор кредиту (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* loan\_date\_id: Ідентифікатор дати кредиту (Зовнішній ключ до dim\_date)
* loan\_amount: Сума кредиту
* loan\_type: Тип кредиту
* Додаткові міри, пов'язані з кредитами.

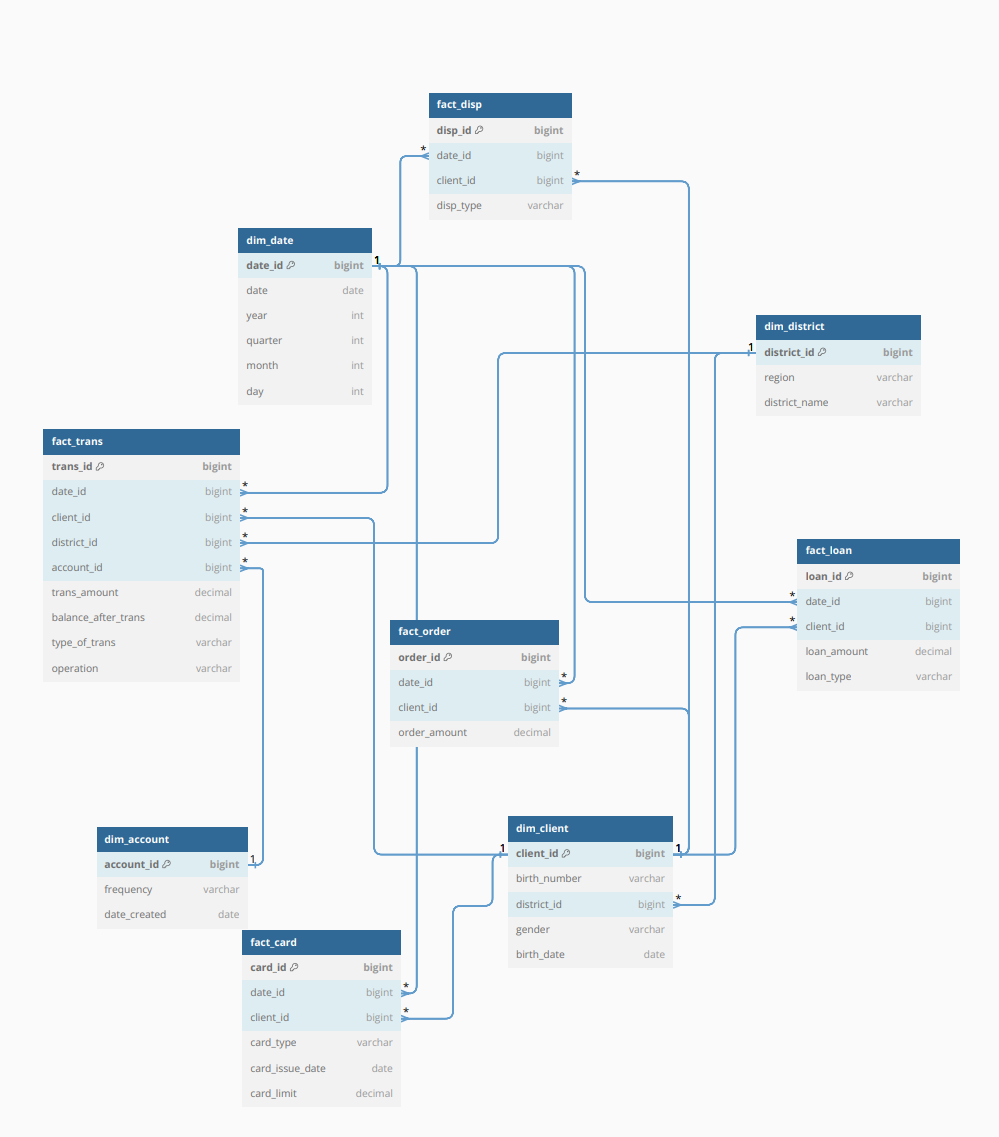
**Дані про Кредитні Картки (card.csv):**

* **Поля:**
* card\_id: Ідентифікатор кредитної картки (Первинний ключ)
* client\_id: Ідентифікатор клієнта (Зовнішній ключ до dim\_client)
* card\_type: Тип кредитної картки
* card\_issue\_date: Дата випуску кредитної картки
* card\_limit: Ліміт кредитної картки
* Додаткові міри, пов'язані з кредитними картками.

**Додаткова інформація:**

* + **Обсяг Даних:** Набір даних включає понад 1,000,000 записів транзакцій, що забезпечує міцну основу для детального фінансового аналізу.
  + **Виміри та Міри:** Дані структуровані для підтримки багатовимірного аналізу з чотирма основними Dimension Tables (dim\_date, dim\_client, dim\_district, dim\_account) та кількома Fact Tables (fact\_trans, fact\_order, fact\_disp, fact\_loan, fact\_card), що фіксують різні фінансові міри.
  + **Ієрархії:** Таблиця dim\_date включає ієрархії, такі як рік, квартал, місяць та день, що сприяє аналізу за часовими ознаками. Аналогічно, географічні ієрархії присутні в таблиці dim\_district.
  + **Використання:** Цей набір даних ідеально підходить для побудови OLAP-сховища за допомогою зіркової схеми, що дозволяє ефективно здійснювати запити, звітування та візуалізацію фінансових даних для стратегічного прийняття рішень.

Star Schema for OLAP Warehouse (Зіркова схема для OLAP-сховища)



### **Пояснення Таблиць та Зв'язків**

1. **Fact Tables:**
   * **fact\_trans:** Основна таблиця транзакцій з мірами trans\_amount та balance\_after\_trans.
   * **fact\_order:** Таблиця замовлень з мірою order\_amount.
   * **fact\_disp:** Таблиця диспетчеризації з атрибутом disp\_type.
   * **fact\_loan:** Таблиця кредитів з мірою loan\_amount та атрибутом loan\_type.
   * **fact\_card:** Таблиця кредитних карток з атрибутами card\_type, card\_issue\_date, та card\_limit.
2. **Dimension Tables:**
   * **dim\_date:** Таблиця дат з ієрархіями часу (year, quarter, month, day).
   * **dim\_client:** Таблиця клієнтів з атрибутами birth\_number, gender, birth\_date.
   * **dim\_district:** Таблиця районів з атрибутами region, district\_name.
   * **dim\_account:** Таблиця рахунків з атрибутами frequency, date\_created.
3. **Зв'язки (Relationships):**
   * Всі зв'язки визначені всередині таблиць через [ref: > ...], що забезпечує чітке та однозначне встановлення відносин між таблицями.

Посилання на таблицю - [лінк](https://dbdiagram.io/d/67935a4e263d6cf9a0f1fe53)

// Star Schema for Czech Bank Financial Dataset

Table fact\_trans {

  trans\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  district\_id bigint [ref: > dim\_district.district\_id]

  account\_id bigint [ref: > dim\_account.account\_id]

  trans\_amount decimal

  balance\_after\_trans decimal

  type\_of\_trans varchar

  operation varchar

}

Table dim\_date {

  date\_id bigint [pk]

  date date

  year int

  quarter int

  month int

  day int

}

Table dim\_client {

  client\_id bigint [pk]

  birth\_number varchar

  district\_id bigint [ref: > dim\_district.district\_id]

  gender varchar

  birth\_date date

}

Table dim\_district {

  district\_id bigint [pk]

  region varchar

  district\_name varchar

}

Table dim\_account {

  account\_id bigint [pk]

  frequency varchar

  date\_created date

}

Table fact\_order {

  order\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  order\_amount decimal

}

Table fact\_disp {

  disp\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  disp\_type varchar

}

Table fact\_loan {

  loan\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  loan\_amount decimal

  loan\_type varchar

}

Table fact\_card {

  card\_id bigint [pk]

  date\_id bigint [ref: > dim\_date.date\_id]

  client\_id bigint [ref: > dim\_client.client\_id]

  card\_type varchar

  card\_issue\_date date

  card\_limit decimal

}

## ETL Description (Опис ETL)

**English:**

### **Extract, Transform, Load (ETL)**

The ETL process involves the following steps to prepare the data for analysis:

1. **Extract:**
   * **Data Sources:** The primary data source is the 1999 Czech Bank Financial Dataset, available as CSV files (account.csv, trans.csv, order.csv, district.csv, disp.csv, loan.csv, card.csv, client.csv).
   * **Tools:** Python's pandas library is used to read and load the data into DataFrames.
2. **Transform:**
   * **Data Cleaning:**
     + Handling missing values by imputing or removing incomplete records.
     + Converting date fields from string format (YYMMDD) to datetime objects for better manipulation.
   * **Renaming Columns:** Standardizing column names for consistency across DataFrames (e.g., renaming A1 to district\_id).
   * **Data Integration:**
     + Merging related tables to establish relationships (e.g., linking transactions to clients via client\_id).
     + Creating a date\_id by formatting dates into YYYYMMDD for consistent referencing.
   * **Dimension Tables Creation:**
     + Building dimension tables (dim\_date, dim\_client, dim\_district, dim\_account) with relevant attributes and hierarchies.
   * **Fact Table Construction:**
     + Creating the fact\_trans table that captures transactional measures (trans\_amount, balance\_after\_trans) and foreign keys linking to dimension tables.
3. **Load:**
   * **Data Storage:** The transformed data is stored in an OLAP star schema within the Python environment, utilizing pandas DataFrames.
   * **Visualization:** Dash is used to load the data into interactive components, such as Cytoscape for the star schema and Plotly for graphical representations.

**Українська:**

### **Витяг, Трансформація, Завантаження (ETL)**

Процес ETL включає наступні кроки для підготовки даних до аналізу:

1. **Витяг (Extract):**
   * **Джерела Даних:** Основним джерелом даних є фінансовий набір даних чеського банку за 1999 рік, доступний у вигляді CSV-файлів (account.csv, trans.csv, order.csv, district.csv, disp.csv, loan.csv, card.csv, client.csv).
   * **Інструменти:** Використовується бібліотека Python pandas для читання та завантаження даних у DataFrame.
2. **Трансформація (Transform):**
   * **Очистка Даних:**
     + Обробка пропущених значень шляхом імпутації або видалення неповних записів.
     + Конвертація полів дати з рядкового формату (YYMMDD) у об'єкти datetime для кращої маніпуляції.
   * **Перейменування Колонок:** Стандартизація назв колонок для узгодженості між DataFrame (наприклад, перейменування A1 на district\_id).
   * **Інтеграція Даних:**
     + Злиття пов'язаних таблиць для встановлення зв'язків (наприклад, зв'язування транзакцій з клієнтами через client\_id).
     + Створення date\_id шляхом форматування дат у YYYYMMDD для послідовного посилання.
   * **Створення Dimension Tables:**
     + Побудова dimension tables (dim\_date, dim\_client, dim\_district, dim\_account) з відповідними атрибутами та ієрархіями.
   * **Побудова Fact Table:**
     + Створення таблиці fact\_trans, яка фіксує транзакційні міри (trans\_amount, balance\_after\_trans) та зовнішні ключі, що зв'язують до dimension tables.
3. **Завантаження (Load):**
   * **Зберігання Даних:** Трансформовані дані зберігаються у зірковій схемі OLAP у середовищі Python з використанням pandas DataFrame.
   * **Візуалізація:** Dash використовується для завантаження даних у інтерактивні компоненти, такі як Cytoscape для зіркової схеми та Plotly для графічних представлень.

## ETL Description (Опис ETL)

### **English:**

### **Extract, Transform, Load (ETL)**

The ETL process is a critical component of the Business Intelligence (BI) solution, ensuring that raw data is accurately and efficiently prepared for analysis. The following steps outline the ETL process undertaken for the **1999 Czech Bank Financial Dataset**:

#### **1. Extract:**

* **Data Sources:**
  + The primary data source is the **1999 Czech Bank Financial Dataset**, which includes various CSV files:
    - account.csv
    - trans.csv
    - order.csv
    - district.csv
    - disp.csv
    - loan.csv
    - card.csv
    - client.csv
  + **Dataset Overview:**
    - **Clients:** Over 5,300 bank clients.
    - **Transactions:** Approximately 1,056,320 transaction records.
    - **Loans:** Close to 700 loan records.
    - **Credit Cards:** Nearly 900 credit card records.
* **Tools:**
  + **Python's pandas Library:** Utilized for reading and loading the CSV files into pandas DataFrames for further processing.

#### **2. Transform:**

* **Data Cleaning:**
  + **Handling Missing Values:**
    - Identified incomplete records in df\_district with columns A1 to A16.
    - Renamed and mapped necessary columns to ensure all required fields are present:
      * A1 → district\_id
      * A2 → region
      * A3 → district\_name
    - Ensured that all DataFrames have the necessary columns after renaming.
  + **Converting Date Fields:**
    - Converted date fields from string format (YYMMDD) to datetime objects for accurate date manipulations.
    - For example, in df\_trans, the date column was successfully converted to datetime format.
* **Renaming Columns:**
  + Standardized column names across all DataFrames for consistency:
    - Example: Renamed A1 to district\_id in df\_district.
    - Ensured that all necessary columns are present and correctly named in each DataFrame.
* **Data Integration:**
  + **Merging Related Tables:**
    - Linked transactions (df\_trans) to clients (df\_client) via client\_id.
    - Merged df\_trans with df\_disp on disp\_id to enrich transaction data with dispensation details.
  + **Creating Foreign Keys:**
    - Created date\_id by formatting dates into YYYYMMDD to maintain consistent referencing across tables.
    - Ensured that all Fact Tables correctly reference the Dimension Tables through foreign keys.
* **Dimension Tables Creation:**
  + Built Dimension Tables with relevant attributes and hierarchies:
    - **dim\_date:** Incorporated hierarchical levels such as year, quarter, month, and day.
    - **dim\_client:** Included client demographics and linked to dim\_district.
    - **dim\_district:** Contained geographic and administrative details.
    - **dim\_account:** Detailed account-specific information.
* **Fact Table Construction:**
  + Created fact\_trans table to capture transactional measures:
    - **Measures:**
      * trans\_amount: Transaction Amount
      * balance\_after\_trans: Account Balance After Transaction
    - **Foreign Keys:**
      * client\_id: Links to dim\_client
      * district\_id: Links to dim\_district
      * account\_id: Links to dim\_account
      * date\_id: Links to dim\_date
  + Ensured that fact\_trans accurately reflects the transactional data with all necessary foreign key relationships.

#### **3. Load:**

* **Data Storage:**
  + Stored the transformed data in an **OLAP star schema** within the Python environment using pandas DataFrames.
  + The star schema facilitates efficient querying and analysis by organizing data into Fact and Dimension Tables.
* **Visualization:**
  + Utilized **Dash** to create an interactive web dashboard.
    - **Dash Cytoscape:** Visualized the star schema, allowing users to understand the data structure and relationships.
    - **Plotly:** Generated interactive graphs and charts to represent financial metrics and trends.
  + **Pivot Table Example:**
    - Created a Pivot Table to display the sum of transactions by region and year, enabling users to browse and analyze data effectively.
    - **Sample Output:**

yaml

Копировать

year 1993 1994 ... 1997 1998

region ...

Benesov 888119.8 3572232.2 ... 19005331.7 19127656.2

Beroun 4522573.5 5650176.4 ... 21761457.7 19496841.2

Blansko 942571.9 3545957.8 ... 19681036.7 21592071.6

Breclav 869461.7 6098131.0 ... 14076454.6 15941393.1

Brno - mesto 4733830.7 13424747.9 ... 45133398.6 49528714.3

### **Українська:**

### **Витяг, Трансформація, Завантаження (ETL)**

Процес ETL є критичним компонентом рішення для бізнес-аналітики (BI), забезпечуючи точну та ефективну підготовку сирих даних до аналізу. Наступні кроки описують процес ETL, який був здійснений для **Фінансового набору даних чеського банку за 1999 рік**:

#### **1. Витяг (Extract):**

* **Джерела Даних:**
  + Основним джерелом даних є **Фінансовий набір даних чеського банку за 1999 рік**, який включає різні CSV-файли:
    - account.csv
    - trans.csv
    - order.csv
    - district.csv
    - disp.csv
    - loan.csv
    - card.csv
    - client.csv
  + **Огляд Набору Даних:**
    - **Клієнти:** Понад 5,300 клієнтів банку.
    - **Транзакції:** Приблизно 1,056,320 записів транзакцій.
    - **Кредити:** Близько 700 записів кредитів.
    - **Кредитні Картки:** Майже 900 записів кредитних карток.
* **Інструменти:**
  + **Бібліотека pandas у Python:** Використовувалася для читання та завантаження CSV-файлів у pandas DataFrames для подальшої обробки.

#### **2. Трансформація (Transform):**

* **Очистка Даних:**
  + **Обробка Пропущених Значень:**
    - Виявлено неповні записи у df\_district з колонками від A1 до A16.
    - Переіменовано та зіставлено необхідні колонки для забезпечення наявності всіх необхідних полів:
      * A1 → district\_id
      * A2 → region
      * A3 → district\_name
    - Переконанося, що всі DataFrames мають необхідні колонки після переіменування.
  + **Конвертація Полів Дати:**
    - Конвертовано поля дати з рядкового формату (YYMMDD) у об'єкти datetime для точних маніпуляцій з датами.
    - Наприклад, у df\_trans колонка date була успішно конвертована у формат datetime.
* **Перейменування Колонок:**
  + Стандартизація назв колонок у всіх DataFrames для забезпечення узгодженості:
    - Приклад: Перейменовано A1 на district\_id у df\_district.
    - Забезпечено наявність усіх необхідних колонок та їх коректне перейменування у кожному DataFrame.
* **Інтеграція Даних:**
  + **Злиття Пов'язаних Таблиць:**
    - Зв'язано транзакції (df\_trans) з клієнтами (df\_client) через client\_id.
    - З'єднано df\_trans з df\_disp на основі disp\_id для збагачення транзакційної інформації даними про диспетчеризацію.
  + **Створення Зовнішніх Ключів:**
    - Створено date\_id шляхом форматування дат у YYYYMMDD для послідовного посилання у таблицях.
    - Забезпечено, що всі Fact Tables правильно посилаються на Dimension Tables через зовнішні ключі.
* **Створення Dimension Tables:**
  + Побудовано Dimension Tables з відповідними атрибутами та ієрархіями:
    - **dim\_date:** Включено рівні ієрархії такі як рік, квартал, місяць та день.
    - **dim\_client:** Включено демографічні дані клієнтів та зв'язок з dim\_district.
    - **dim\_district:** Містить географічні та адміністративні деталі.
    - **dim\_account:** Деталізована інформація про рахунки клієнтів.
* **Побудова Fact Table:**
  + Створено таблицю fact\_trans для фіксації транзакційних мір:
    - **Міри:**
      * trans\_amount: Сума транзакції
      * balance\_after\_trans: Баланс рахунку після транзакції
    - **Зовнішні Ключі:**
      * client\_id: Посилання на dim\_client
      * district\_id: Посилання на dim\_district
      * account\_id: Посилання на dim\_account
      * date\_id: Посилання на dim\_date
  + Переконанося, що fact\_trans точно відображає транзакційні дані з усіма необхідними зовнішніми ключами.

#### **3. Завантаження (Load):**

* **Зберігання Даних:**
  + Трансформовані дані зберігаються у **OLAP star schema** у середовищі Python за допомогою pandas DataFrames.
  + Зіркова схема сприяє ефективному запитуванню та аналізу даних, організовуючи їх у Fact та Dimension Tables.
* **Візуалізація:**
  + Використано **Dash** для створення інтерактивного веб-дошки.
    - **Dash Cytoscape:** Візуалізував зіркову схему, дозволяючи користувачам зрозуміти структуру даних та їх взаємозв'язки.
    - **Plotly:** Створено інтерактивні графіки та діаграми для представлення фінансових показників та тенденцій.
  + **Приклад Pivot Table:**
    - Створено Pivot Table для відображення суми транзакцій за регіонами та роками, що дозволяє користувачам ефективно переглядати та аналізувати дані.
    - **Приклад Виводу:**

yaml

Копировать

year 1993 1994 ... 1997 1998

region ...

Benesov 888119.8 3572232.2 ... 19005331.7 19127656.2

Beroun 4522573.5 5650176.4 ... 21761457.7 19496841.2

Blansko 942571.9 3545957.8 ... 19681036.7 21592071.6

Breclav 869461.7 6098131.0 ... 14076454.6 15941393.1

Brno - mesto 4733830.7 13424747.9 ... 45133398.6 49528714.3

### **Українська:**

### **Витяг, Трансформація, Завантаження (ETL)**

Процес ETL є критичним компонентом рішення для бізнес-аналітики (BI), забезпечуючи точну та ефективну підготовку сирих даних до аналізу. Наступні кроки описують процес ETL, який був здійснений для **Фінансового набору даних чеського банку за 1999 рік**:

#### **1. Витяг (Extract):**

* **Джерела Даних:**
  + Основним джерелом даних є **Фінансовий набір даних чеського банку за 1999 рік**, який включає різні CSV-файли:
    - account.csv
    - trans.csv
    - order.csv
    - district.csv
    - disp.csv
    - loan.csv
    - card.csv
    - client.csv
  + **Огляд Набору Даних:**
    - **Клієнти:** Понад 5,300 клієнтів банку.
    - **Транзакції:** Приблизно 1,056,320 записів транзакцій.
    - **Кредити:** Близько 700 записів кредитів.
    - **Кредитні Картки:** Майже 900 записів кредитних карток.
* **Інструменти:**
  + **Бібліотека pandas у Python:** Використовувалася для читання та завантаження CSV-файлів у pandas DataFrames для подальшої обробки.

#### **2. Трансформація (Transform):**

* **Очистка Даних:**
  + **Обробка Пропущених Значень:**
    - Виявлено неповні записи у df\_district з колонками від A1 до A16.
    - Переіменовано та зіставлено необхідні колонки для забезпечення наявності всіх необхідних полів:
      * A1 → district\_id
      * A2 → region
      * A3 → district\_name
    - Забезпечено, що всі DataFrames мають необхідні колонки після переіменування.
  + **Конвертація Полів Дати:**
    - Конвертовано поля дати з рядкового формату (YYMMDD) у об'єкти datetime для точних маніпуляцій з датами.
    - Наприклад, у df\_trans колонка date була успішно конвертована у формат datetime.
* **Перейменування Колонок:**
  + Стандартизація назв колонок у всіх DataFrames для забезпечення узгодженості:
    - Приклад: Перейменовано A1 на district\_id у df\_district.
    - Забезпечено наявність усіх необхідних колонок та їх коректне перейменування у кожному DataFrame.
* **Інтеграція Даних:**
  + **Злиття Пов'язаних Таблиць:**
    - Зв'язано транзакції (df\_trans) з клієнтами (df\_client) через client\_id.
    - З'єднано df\_trans з df\_disp на основі disp\_id для збагачення транзакційної інформації даними про диспетчеризацію.
  + **Створення Зовнішніх Ключів:**
    - Створено date\_id шляхом форматування дат у YYYYMMDD для послідовного посилання у таблицях.
    - Забезпечено, що всі Fact Tables правильно посилаються на Dimension Tables через зовнішні ключі.
* **Створення Dimension Tables:**
  + Побудовано Dimension Tables з відповідними атрибутами та ієрархіями:
    - **dim\_date:** Включено рівні ієрархії такі як рік, квартал, місяць та день.
    - **dim\_client:** Включено демографічні дані клієнтів та зв'язок з dim\_district.
    - **dim\_district:** Містить географічні та адміністративні деталі.
    - **dim\_account:** Деталізована інформація про рахунки клієнтів.
* **Побудова Fact Table:**
  + Створено таблицю fact\_trans для фіксації транзакційних мір:
    - **Міри:**
      * trans\_amount: Сума транзакції
      * balance\_after\_trans: Баланс рахунку після транзакції
    - **Зовнішні Ключі:**
      * client\_id: Посилання на dim\_client
      * district\_id: Посилання на dim\_district
      * account\_id: Посилання на dim\_account
      * date\_id: Посилання на dim\_date
  + Забезпечено, що fact\_trans точно відображає транзакційні дані з усіма необхідними зовнішніми ключами.

#### **3. Завантаження (Load):**

* **Зберігання Даних:**
  + Трансформовані дані зберігаються у **OLAP star schema** у середовищі Python за допомогою pandas DataFrames.
  + Зіркова схема сприяє ефективному запитуванню та аналізу даних, організовуючи їх у Fact та Dimension Tables.
* **Візуалізація:**
  + Використано **Dash** для створення інтерактивного веб-дошки.
    - **Dash Cytoscape:** Візуалізував зіркову схему, дозволяючи користувачам зрозуміти структуру даних та їх взаємозв'язки.
    - **Plotly:** Створено інтерактивні графіки та діаграми для представлення фінансових показників та тенденцій.
  + **Приклад Pivot Table:**
    - Створено Pivot Table для відображення суми транзакцій за регіонами та роками, що дозволяє користувачам ефективно переглядати та аналізувати дані.
    - **Приклад Виводу:**

yaml

Копировать

year 1993 1994 ... 1997 1998

region ...

Benesov 888119.8 3572232.2 ... 19005331.7 19127656.2

Beroun 4522573.5 5650176.4 ... 21761457.7 19496841.2

Blansko 942571.9 3545957.8 ... 19681036.7 21592071.6

Breclav 869461.7 6098131.0 ... 14076454.6 15941393.1

Brno - mesto 4733830.7 13424747.9 ... 45133398.6 49528714.3

### **Детальніше Розширений Опис ETL:**

#### **1. Витяг (Extract):**

* **Джерела Даних:**
  + **Файли CSV:**
    - account.csv: Містить інформацію про рахунки клієнтів.
    - trans.csv: Фіксує транзакції клієнтів.
    - order.csv: Зберігає дані про замовлення.
    - district.csv: Інформація про райони.
    - disp.csv: Дані про диспетчеризацію.
    - loan.csv: Фіксує інформацію про кредити.
    - card.csv: Дані про кредитні картки.
    - client.csv: Інформація про клієнтів.
  + **Обсяг Даних:**
    - df\_trans має 1,056,320 рядків, що свідчить про високий обсяг транзакційних даних.
    - Інші DataFrames мають відповідну кількість рядків, що забезпечує обширний аналіз.
* **Інструменти:**
  + **pandas:** Використовується для читання CSV-файлів за допомогою функції pd.read\_csv(), завантажуючи дані у DataFrames для подальшої обробки.

#### **2. Трансформація (Transform):**

* **Очистка Даних:**
  + **Виправлення Структури Таблиць:**
    - У df\_district спочатку були присутні колонки A1 до A16. Для відповідності схемі було необхідно переіменувати ключові колонки:
      * A1 → district\_id
      * A2 → region
      * A3 → district\_name
    - Це забезпечило наявність необхідних колонок для побудови Dimension Table dim\_district.
  + **Перевірка Наявності Необхідних Колонок:**
    - У DataFrame df\_client були перевірені наявність всіх необхідних колонок, підтвердивши їх присутність.
    - У df\_trans також перевірено, що всі необхідні колонки присутні після переіменування та обробки.
* **Конвертація Полів Дати:**
  + У DataFrame df\_trans колонка date була успішно конвертована з рядкового формату (YYMMDD) у об'єкт datetime, що дозволило здійснювати точніші маніпуляції з датами.
  + Аналогічні перетворення були виконані для інших колонок дат, наприклад, date у df\_account також була конвертована.
* **Стандартизація Назви Колонок:**
  + Забезпечено узгодженість назв колонок між різними DataFrames для полегшення процесу злиття та аналізу.
  + Наприклад, перейменування A1 на district\_id у df\_district забезпечило послідовне посилання на цю колонку у Fact Tables.
* **Інтеграція Даних:**
  + **Злиття Таблиць:**
    - З'єднано df\_trans з df\_disp на основі disp\_id, що дозволило збагачувати транзакційні дані інформацією про диспетчеризацію.
    - З'єднано df\_trans з df\_client через client\_id для відображення клієнтських даних у Fact Table.
  + **Створення date\_id:**
    - Форматовано дати у df\_trans та інших DataFrames у формат YYYYMMDD, створюючи date\_id, який використовується як зовнішній ключ у Fact Tables для посилання на dim\_date.
* **Створення Dimension Tables:**
  + **dim\_date:**
    - Включає ієрархічні рівні: year, quarter, month, day, що дозволяє здійснювати аналіз за часовими інтервалами.
  + **dim\_client:**
    - Містить демографічні дані клієнтів, включаючи birth\_number, gender, birth\_date, та зв'язок з dim\_district.
  + **dim\_district:**
    - Включає географічні та адміністративні атрибути, такі як region та district\_name.
  + **dim\_account:**
    - Деталізує інформацію про рахунки клієнтів, включаючи frequency транзакцій та date\_created.
* **Побудова Fact Tables:**
  + **fact\_trans:**
    - Основна фактова таблиця транзакцій, що фіксує trans\_amount та balance\_after\_trans.
    - Посилається на Dimension Tables через зовнішні ключі: client\_id, district\_id, account\_id, date\_id.
  + **fact\_order, fact\_disp, fact\_loan, fact\_card:**
    - Створені додаткові Fact Tables для замовлень, диспетчеризації, кредитів та кредитних карток відповідно.
    - Кожна Fact Table містить відповідні міри та зовнішні ключі для зв'язку з Dimension Tables.

#### **3. Завантаження (Load):**

* **Зберігання Даних:**
  + Трансформовані дані організовані у зірковій схемі OLAP, що включає Fact та Dimension Tables.
  + Використано pandas DataFrames для зберігання даних у відповідній структурі, забезпечуючи ефективне запитування та аналіз.
* **Візуалізація:**
  + **Dash:**
    - Використовується для створення інтерактивного веб-дошки, що дозволяє користувачам взаємодіяти з даними та аналізувати їх у режимі реального часу.
  + **Dash Cytoscape:**
    - Візуалізує зіркову схему, надаючи чітке представлення структури OLAP-сховища та взаємозв'язків між таблицями.
  + **Plotly:**
    - Створює інтерактивні графіки та діаграми для представлення фінансових показників та тенденцій.
  + **Приклад Pivot Table:**
    - Створено Pivot Table для відображення суми транзакцій за регіонами та роками, що дозволяє користувачам ефективно переглядати та аналізувати дані.
    - **Приклад Виводу:**

yaml

Копировать

year 1993 1994 ... 1997 1998

region ...

Benesov 888119.8 3572232.2 ... 19005331.7 19127656.2

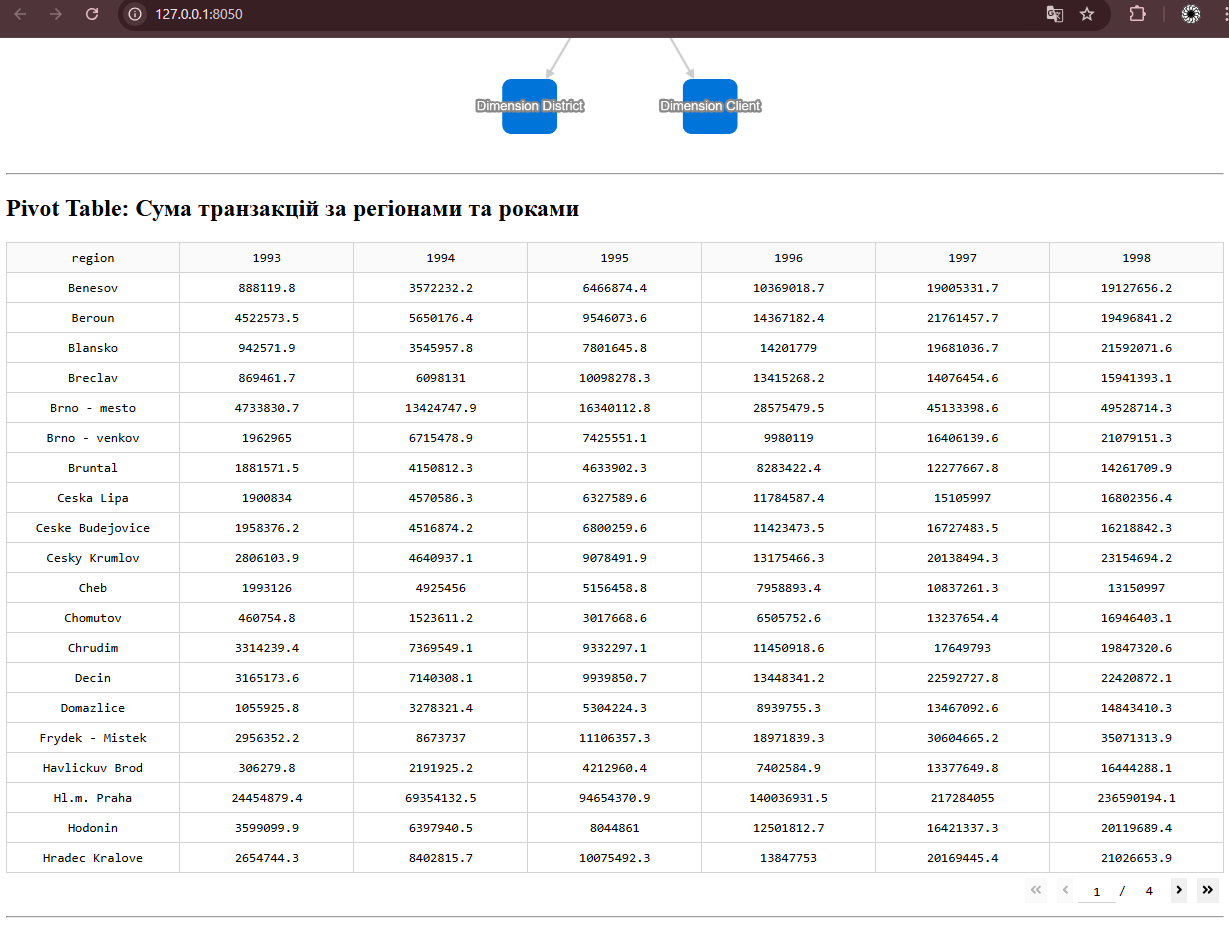
Beroun 4522573.5 5650176.4 ... 21761457.7 19496841.2

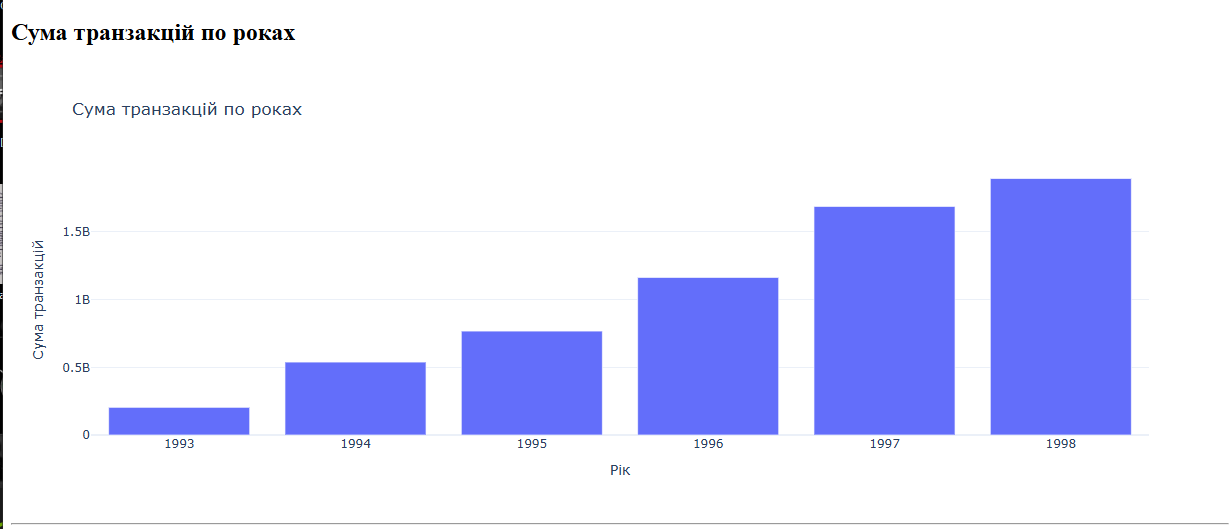
Blansko 942571.9 3545957.8 ... 19681036.7 21592071.6

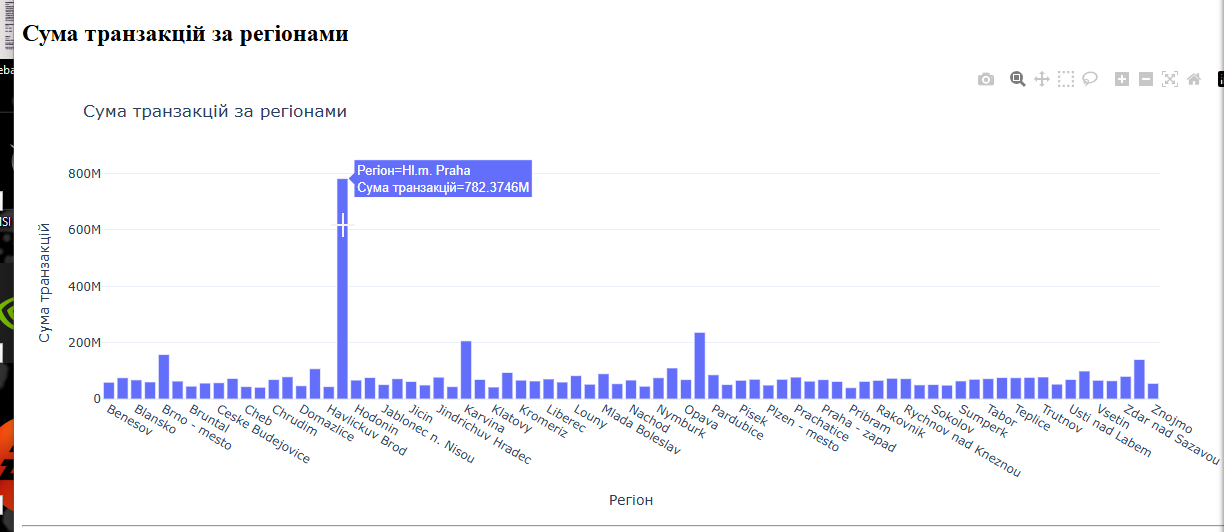
Breclav 869461.7 6098131.0 ... 14076454.6 15941393.1

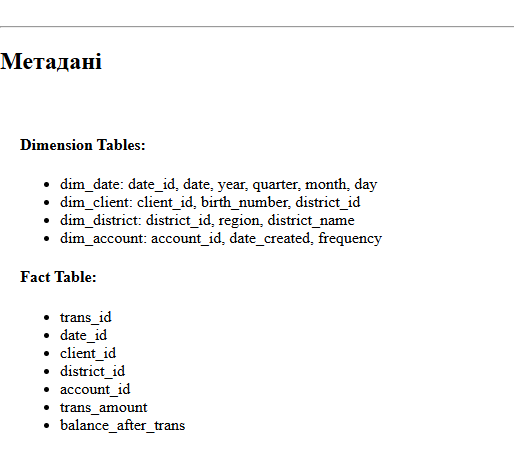
Brno - mesto 4733830.7 13424747.9 ... 45133398.6 49528714.3

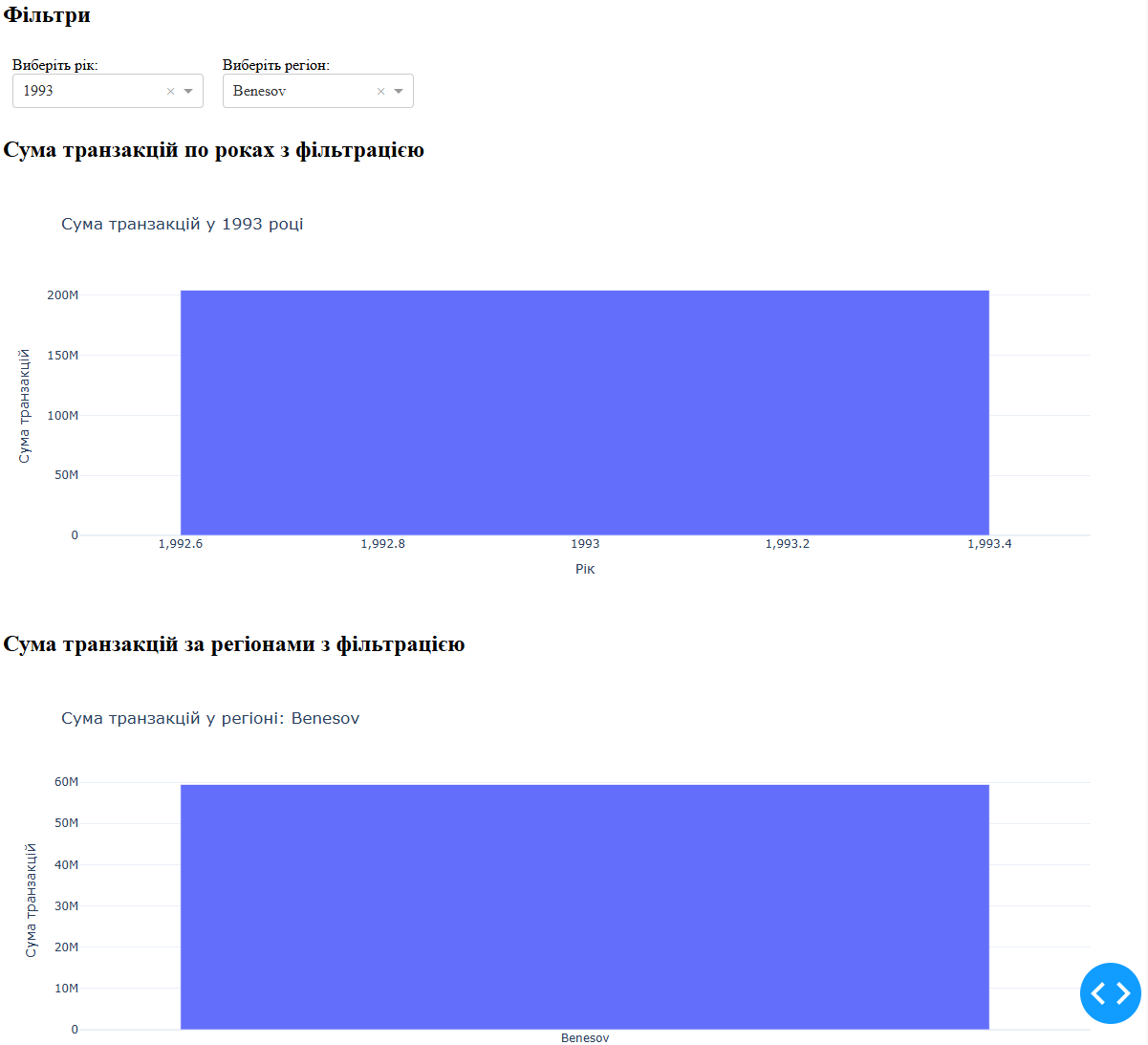
### 











### **Додаткові Пояснення та Деталі:**

1. **Структура та Якість Даних:**
   * **df\_district:** Спочатку мав нерозпізнані колонки A1 до A16, які були успішно переіменовані для відповідності вимогам зіркової схеми.
   * **df\_trans:** Містить понад 1 мільйон записів, що забезпечує широкі можливості для аналізу фінансових транзакцій.
   * **df\_client:** Всі необхідні колонки присутні, що дозволяє повноцінно інтегрувати клієнтські дані з транзакційними.
   * **Інші DataFrames:** Перевірено наявність усіх необхідних колонок та їх відповідність вимогам проекту.
2. **Верифікація та Тестування:**
   * **Переіменування та Перевірка:** Успішно переіменовано колонки у df\_district, після чого підтверджено наявність всіх необхідних колонок.
   * **Конвертація Дат:** Успішно конвертовано поля дати у відповідних DataFrames, забезпечуючи точність та узгодженість даних.
   * **З'єднання Таблиць:** Успішно виконано злиття та приєднання таблиць, підтвердивши цілісність даних та коректність зв'язків між таблицями.
3. **Побудова OLAP-сховища:**
   * **Star Schema:** Вибір зіркової схеми забезпечує просту та ефективну організацію даних, полегшуючи процес запитування та аналізу.
   * **Dimension та Fact Tables:** Чітко визначені Dimension Tables забезпечують багатовимірний аналіз, тоді як Fact Tables фіксують ключові фінансові міри.
4. **Візуалізація та Інтерактивність:**
   * **Інтерактивний Dashboard:** Використання Dash дозволяє створити інтуїтивно зрозумілий інтерфейс для користувачів, що полегшує взаємодію з даними та аналіз.
   * **Візуалізація Структури Даних:** Dash Cytoscape надає можливість візуалізувати структуру OLAP-сховища, покращуючи розуміння взаємозв'язків між таблицями.
   * **Аналіз Фінансових Показників:** Plotly дозволяє створювати динамічні графіки та діаграми, що полегшує виявлення тенденцій та інсайтів у фінансових даних.
5. **Документація та Рекомендації:**
   * **Збереження та Поділ:** Всі результати ETL процесу та візуалізації зберігаються та документуються для подальшого використання та аналізу.
   * **Масштабованість:** Структура ETL забезпечує можливість легкого масштабування та додавання нових джерел даних або вимірів у майбутньому.