

# Лабораторна робота №5

Реалізація бізнес-аналітики для системи TechMarket

Виконав: Кіщук Ярослав

# Зміст

<b>1</b>	<b>Вступ</b>	<b>2</b>
<b>2</b>	<b>Мета та завдання</b>	<b>2</b>
<b>3</b>	<b>Архітектура аналітичного рішення</b>	<b>2</b>
3.1	Структура DWH . . . . .	2
3.2	Інтеграція з Metabase . . . . .	3
<b>4</b>	<b>Ключові показники ефективності (KPI)</b>	<b>3</b>
4.1	KPI 1: Динаміка виручки за місяцями (Revenue by Month) . . . . .	3
4.2	KPI 2: Кількість замовлень за регіонами (Orders by Region) . . . . .	4
4.3	KPI 3: Середній чек (Average Order Value) . . . . .	4
4.4	KPI 4: Відсоток маржі (Margin Percentage) . . . . .	4
4.5	KPI 5: Топ-10 товарів за виручкою (Top Products by Revenue) . . . . .	5
<b>5</b>	<b>Параметризація запитів</b>	<b>5</b>
<b>6</b>	<b>Створення дашборда в Metabase</b>	<b>6</b>
6.1	Архітектура дашборда . . . . .	6
6.2	Глобальні фільтри . . . . .	6
6.3	Автоматизація розгортання . . . . .	6
<b>7</b>	<b>Інтеграція з інфраструктурою</b>	<b>7</b>
7.1	Docker Compose . . . . .	7
7.2	Послідовність запуску . . . . .	7
<b>8</b>	<b>Результати та аналіз</b>	<b>7</b>
8.1	Досягнуті результати . . . . .	7
8.2	Аналіз KPI . . . . .	8
8.3	Переваги використання Metabase . . . . .	8
<b>9</b>	<b>Висновки</b>	<b>8</b>

# 1 Вступ

Метою п'ятої лабораторної роботи є розробка системи бізнес-аналітики (BI) для інтернет-магазину TechMarket. У попередніх лабораторних роботах було спроектовано архітектуру баз даних (OLTP та DWH), реалізовано ETL-процес для завантаження даних із операційних баз у аналітичне сховище. Тепер завданням є створення аналітичних запитів для розрахунку ключових показників ефективності (KPI) та побудова інтерактивних дашбордів для візуалізації цих метрик.

У рамках роботи використано:

- **PostgreSQL DWH** — аналітичне сховище даних із зоряною схемою (6 dimension-таблиць та 1 fact-таблиця).
- **SQL** — мова запитів для розрахунку KPI безпосередньо у базі даних.
- **Metabase** — open-source BI-платформа для візуалізації даних та створення дашбордів.
- **Python** — автоматизація налаштування Metabase через API (`scripts/metabase_seed.py`).

## 2 Мета та завдання

**Завдання 1:** Визначити п'ять ключових показників ефективності (KPI) для аналітики продажів TechMarket.

**Завдання 2:** Розробити SQL-запити для розрахунку кожного KPI на основі зоряної схеми DWH.

**Завдання 3:** Реалізувати параметризовані запити з фільтрацією за датами та регіонами.

**Завдання 4:** Інтегрувати Metabase з PostgreSQL DWH для візуалізації даних.

**Завдання 5:** Створити інтерактивний дашборд із п'ятьма візуалізаціями та глобальними фільтрами.

**Завдання 6:** Автоматизувати розгортання BI-інфраструктури через Python-скрипт.

## 3 Архітектура аналітичного рішення

### 3.1 Структура DWH

Аналітичне сховище даних побудовано за принципом зоряної схеми (star schema), що забезпечує оптимальну продуктивність аналітичних запитів. Центральним елементом є таблиця фактів `fact_sales`, яка містить метрики продажів та зовнішні ключі до вимірів.

Табл. 1: Структура DWH схеми

Таблиця	Опис
<code>fact_sales</code>	Факт продажів: <code>order_id</code> , <code>date_key</code> , <code>product_key</code> , <code>customer_key</code> , <code>employee_key</code> , <code>region_key</code> , <code>quantity</code> , <code>revenue</code> , <code>discount_amount</code> , <code>cost</code> , <code>margin</code>
<code>dim_date</code>	Календар: <code>date_key</code> (YYYYMMDD), <code>date</code> , <code>year</code> , <code>quarter</code> , <code>month</code> , <code>day</code> , <code>day_of_week</code> , <code>is_weekend</code>
<code>dim_product</code>	Товари: <code>product_key</code> , <code>product_id</code> , <code>name</code> , <code>sku</code> , <code>category_key</code>

dim_customer	Клієнти: customer_key, customer_id, first_name, last_name, email, region_key
dim_employee	Менеджери: employee_key, employee_id, first_name, last_name, email, region_key
dim_region	Регіони: region_key, region_id, name, code
dim_category	Категорії: category_key, category_id, name, parent_category_key

---

## 3.2 Інтеграція з Metabase

Metabase розгорнуто як Docker-контейнер (metabase/metabase:v0.48.6) і підключено до PostgreSQL DWH через нативний драйвер. Конфігурація здійснюється автоматично за допомогою Python-скрипта metabase\_seed.py, який виконує:

- Автентифікацію в Metabase API
- Створення підключення до DWH
- Генерацію SQL-запитів із template tags для параметризації
- Створення Saved Questions (збережених запитів) для кожного KPI
- Побудову дашборда з розміщенням візуалізацій та налаштуванням фільтрів

## 4 Ключові показники ефективності (KPI)

Для аналізу ефективності продажів системи TechMarket визначено п'ять KPI, які покривають різні аспекти бізнесу: тренд виручки, географічний розподіл замовлень, середній чек, маржинальність та популярність товарів.

### 4.1 KPI 1: Динаміка виручки за місяцями (Revenue by Month)

**Бізнес-питання:** Як змінюється виручка компанії в часі? Які місяці є найбільш прибутковими?

**Опис:** Цей показник відображає сумарну виручку, знижки та маржу за кожен місяць. Дозволяє виявити сезонність, тренди зростання або падіння продажів.

**SQL-запит:**

Лістинг 1: Динаміка виручки за місяцями

```

1 SELECT
2     d.year,
3     d.month,
4     to_char(d.date, 'Mon') AS month_name,
5     SUM(f.revenue)          AS revenue,
6     SUM(f.discount_amount) AS discount,
7     SUM(f.margin)           AS margin
8 FROM fact_sales f
9 JOIN dim_date d ON f.date_key = d.date_key
10 LEFT JOIN dim_region r ON f.region_key = r.region_key
11 WHERE 1=1
12     AND d.date >= '2024-01-01'
13     AND d.date <= '2024-12-31'
14     AND r.name = 'Kyivska'

```

```

15 GROUP BY d.year, d.month, to_char(d.date, 'Mon')
16 ORDER BY d.year, d.month;

```

**Тип візуалізації:** Line chart (лінійний графік) для відображення трендів у часі.

## 4.2 KPI 2: Кількість замовлень за регіонами (Orders by Region)

**Бізнес-питання:** Які регіони генерують найбільше замовлень? Яка географічна структура продажів?

**Опис:** Показник агрегує кількість унікальних замовлень та сумарну виручку в розрізі регіонів. Допомогає виявити найбільш активні ринки збуту.

**SQL-запит:**

Лістинг 2: Замовлення за регіонами

```

1 SELECT
2     r.name AS region_name,
3     COUNT(DISTINCT f.order_id) AS orders_count,
4     SUM(f.revenue) AS revenue
5 FROM fact_sales f
6 JOIN dim_region r ON f.region_key = r.region_key
7 WHERE 1=1
8     AND f.date_key >= 20240101
9     AND f.date_key <= 20241231
10 GROUP BY r.name
11 ORDER BY orders_count DESC;

```

**Тип візуалізації:** Bar chart (стовпчикова діаграма) для порівняння регіонів.

## 4.3 KPI 3: Середній чек (Average Order Value)

**Бізнес-питання:** Скільки в середньому витрачає клієнт за одне замовлення?

**Опис:** Метрика розраховується як відношення сумарної виручки до кількості унікальних замовлень. Є індикатором якості продажів та ефективності маркетингових акцій.

**SQL-запит:**

Лістинг 3: Середній чек

```

1 SELECT
2     SUM(f.revenue) / NULLIF(COUNT(DISTINCT f.order_id), 0)
3     AS avg_order_value
4 FROM fact_sales f
5 LEFT JOIN dim_region r ON f.region_key = r.region_key
6 JOIN dim_date d ON f.date_key = d.date_key
7 WHERE 1=1
8     AND d.date >= '2024-01-01'
9     AND d.date <= '2024-12-31'
10    AND r.name = 'Kyivska';

```

**Тип візуалізації:** Scalar (число) — одне велике значення для швидкого огляду.

## 4.4 KPI 4: Відсоток маржі (Margin Percentage)

**Бізнес-питання:** Яка частка прибутку в загальній виручці?

**Опис:** Показник відображає відношення маржі (різниці між виручкою та собівартістю) до виручки у відсотках. Критичний для оцінки рентабельності бізнесу.

**SQL-запит:**

Лістинг 4: Відсоток маржі

```
1 SELECT
2     (SUM(f.margin) / NULLIF(SUM(f.revenue), 0))::numeric(12,4)
3     AS margin_pct
4 FROM fact_sales f
5 LEFT JOIN dim_region r ON f.region_key = r.region_key
6 JOIN dim_date d ON f.date_key = d.date_key
7 WHERE 1=1
8     AND d.date >= '2024-01-01'
9     AND d.date <= '2024-12-31'
10    AND r.name = 'Kyivska';
```

**Тип візуалізації:** Scalar з форматуванням у відсотки.

## 4.5 KPI 5: Топ-10 товарів за виручкою (Top Products by Revenue)

**Бізнес-питання:** Які товари приносять найбільше доходу? На які товари варто робити акцент у маркетингу?

**Опис:** Рейтинг десяти найприбутковіших товарів із відображенням виручки, проданої кількості та маржі.

**SQL-запит:**

Лістинг 5: Топ-10 товарів

```
1 SELECT
2     p.name AS product_name,
3     SUM(f.revenue) AS revenue,
4     SUM(f.quantity) AS qty,
5     SUM(f.margin) AS margin
6 FROM fact_sales f
7 JOIN dim_product p ON f.product_key = p.product_key
8 LEFT JOIN dim_region r ON f.region_key = r.region_key
9 JOIN dim_date d ON f.date_key = d.date_key
10 WHERE 1=1
11     AND d.date >= '2024-01-01'
12     AND d.date <= '2024-12-31'
13     AND r.name = 'Kyivska'
14 GROUP BY p.name
15 ORDER BY revenue DESC
16 LIMIT 10;
```

**Тип візуалізації:** Bar chart (горизонтальні стовпці) для зручного порівняння товарів.

## 5 Параметризація запитів

Усі п'ять KPI підтримують динамічну фільтрацію через Metabase template tags:

- `{{start_date}}` — початкова дата періоду аналізу (тип: `date`)
- `{{end_date}}` — кінцева дата періоду аналізу (тип: `date`)

- `{{region}}` — назва регіону для фільтрації (тип: `text`, опціональний)

Синтаксис Metabase для опціональних параметрів: `[[AND condition]]`, що дозволяє виключити фільтр, якщо параметр не заданий.

## 6 Створення дашборда в Metabase

### 6.1 Архітектура дашборда

Дашборд “TechMarket KPI Dashboard” складається з п’яти візуалізацій, організованих у grid-layout:

- **Row 0–6:** Revenue by Month (ширина 12 колонок) — займає всю ширину екрана
- **Row 6–12:** Orders by Region (6 колонок), Average Order Value (3 колонки), Margin Percentage (3 колонки)
- **Row 12–18:** Top Products by Revenue (12 колонок)

### 6.2 Глобальні фільтри

На рівні дашборда налаштовано три параметри:

1. **Start Date** — вибір початкової дати (date picker)
2. **End Date** — вибір кінцевої дати (date picker)
3. **Region** — текстове поле для назви регіону (опціонально)

Кожна візуалізація автоматично прив’язана до цих параметрів через `parameter_mappings`, що забезпечує синхронну фільтрацію всіх графіків.

### 6.3 Автоматизація розгортання

Для автоматичного налаштування Metabase створено Python-скрипт `metabase_seed.py`, який:

1. Автентифікується в Metabase API за допомогою `/api/session`
2. Створює або знаходить існуюче підключення до PostgreSQL DWH
3. Генерує п’ять Saved Questions з відповідними SQL-запитами та template tags
4. Створює дашборд та налаштовує глобальні параметри
5. Додає всі візуалізації на дашборд із відповідним layout та mappings

**Приклад запуску:**

```
# Setup environment variables
export METABASE_URL=http://localhost:3000
export METABASE_USER=admin@example.com
export METABASE_PASS=secret

# Run the seed script
python scripts/metabase_seed.py
```

Скрипт також підтримує прапорець `-clear` для видалення попередніх версій дашборда та запитів перед створенням нових.

## 7 Інтеграція з інфраструктурою

### 7.1 Docker Compose

Metabase інтегровано в загальну інфраструктуру TechMarket через `docker-compose.yml`:

```
services:
  metabase:
    image: metabase/metabase:v0.48.6
    container_name: techmarket-metabase
    ports:
      - "3000:3000"
    environment:
      MB_DB_FILE: /metabase-data/metabase.db
    volumes:
      - metabase-data:/metabase-data
    networks:
      - techmarket-net
    depends_on:
      - dwh-db
```

### 7.2 Послідовність запуску

1. Запуск всієї інфраструктури: `docker-compose up -d`
2. Генерація тестових даних в OLTP: `python database/data/generate_test_data.py`
3. Виконання ETL: `python etl/run_etl.py -mode full`
4. Налаштування Metabase: `python scripts/metabase_seed.py`
5. Відкриття дашборда: `http://localhost:3000`

## 8 Результати та аналіз

### 8.1 Досягнуті результати

У рамках лабораторної роботи успішно реалізовано:

- 5 ключових показників ефективності (KPI) для аналізу продажів TechMarket
- SQL-запити з підтримкою параметризації та фільтрації за датами/регіонами
- Інтерактивний дашборд у Metabase із п'ятьма візуалізаціями
- Автоматизацію розгортання BI-інфраструктури через Python API
- Інтеграцію Metabase з PostgreSQL DWH та Docker-екосистемою



## 8.2 Аналіз KPI

Розроблені показники дозволяють відповісти на ключові бізнес-питання:

1. **Revenue by Month** — виявлення трендів та сезонності продажів
2. **Orders by Region** — визначення найбільш активних географічних ринків
3. **Average Order Value** — оцінка ефективності стратегій підвищення чека
4. **Margin Percentage** — контроль рентабельності бізнесу
5. **Top Products** — фокусування маркетингових зусиль на найприбутковіших товарах

## 8.3 Переваги використання Metabase

- **Open-source:** безкоштовна альтернатива Power BI та Tableau
- **Простота інтеграції:** нативна підтримка PostgreSQL без додаткових драйверів
- **REST API:** повна автоматизація через програмний інтерфейс
- **Інтерактивність:** фільтри, drill-down, експорт у CSV/PDF
- **Легке розгортання:** один Docker-контейнер без складних налаштувань

## 9 Висновки

У ході виконання лабораторної роботи №5 було успішно реалізовано повноцінну систему бізнес-аналітики для інтернет-магазину TechMarket. Розроблено п'ять ключових показників ефективності, які покривають різні аспекти діяльності компанії: динаміку виручки, географічний розподіл продажів, середній чек, маржинальність та популярність товарів.

Для кожного KPI створено оптимізовані SQL-запити, які використовують зоряну схему DWH та забезпечують швидке виконання аналітичних операцій. Параметризація запитів дозволяє динамічно фільтрувати дані за датами та регіонами, що підвищує гнучкість аналізу.

Інтеграція з Metabase забезпечила створення інтерактивного дашборда з професійними візуалізаціями. Автоматизація налаштування через Python API дозволяє швидко відтворювати BI-інфраструктуру в різних середовищах (development, staging, production).

Обране рішення (Metabase) виявилось оптимальним для навчальних та малих комерційних проектів завдяки простоті використання, відсутності ліцензійних витрат та повній інтеграції з існуючою Docker-інфраструктурою проекту.

Результати роботи можуть бути використані для прийняття управлінських рішень щодо асортиментної політики, регіональної експансії, ціноутворення та маркетингових стратегій.