

TechMarket: Проєктування та розробка високонавантаженої системи Фінальна презентація проєкту

Ярослав Кіщук

Київський національний університет імені Тараса Шевченка
Факультет комп'ютерних наук та кібернетики

2 грудня 2025 р.

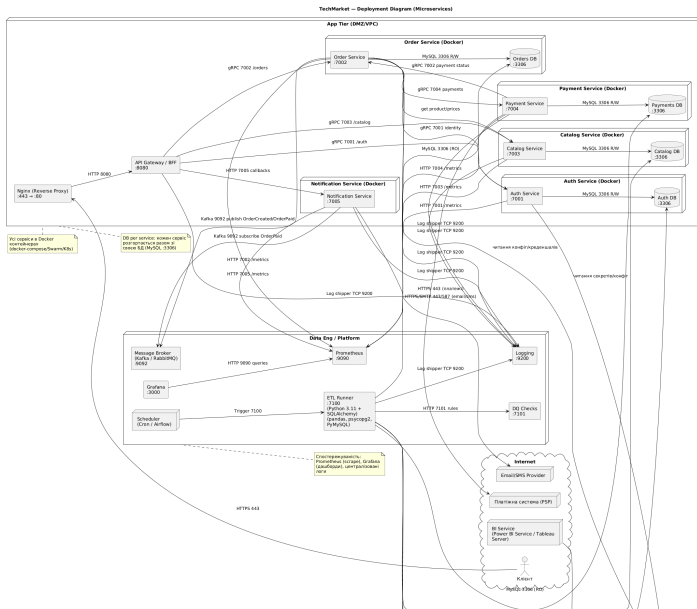
- 1 Вступ
- 2 Архітектура системи
- 3 Модель бази даних (OLTP)
- 4 ETL Процес
- 5 Сховище даних (DWH)
- 6 Лабораторні роботи
- 7 Аналітичні результати
- 8 Висновки

Мета: Розробка масштабованої системи електронної комерції для продажу техніки.

Ключові компоненти:

- Мікросервісна архітектура (Auth, Catalog, Orders, Payments).
- OLTP база даних (MySQL) для транзакційних операцій.
- ETL-пайплайн (Airflow) для обробки даних.
- OLAP сховище (PostgreSQL) для аналітики.
- Оркестрація контейнерів (Kubernetes).

Архітектура розгортання



OLTP Services: Auth & Catalog

ER — Auth Service (MySQL)

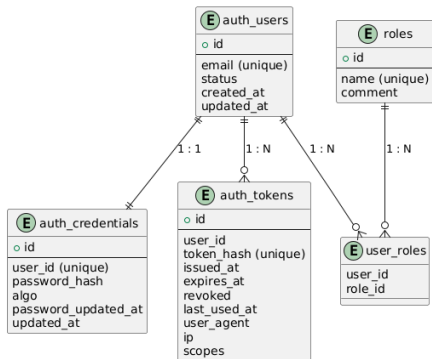


Рис.: Auth Service

ER — Catalog Service (MySQL)

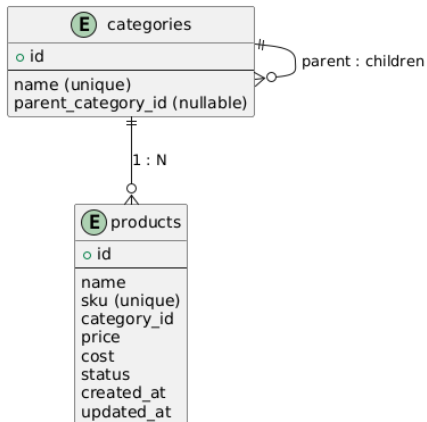
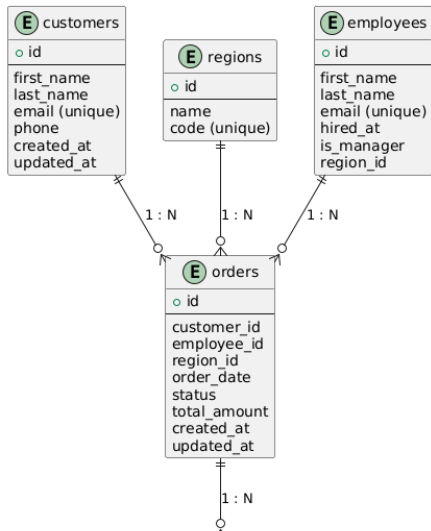


Рис.: Catalog Service

OLTP Services: Orders & Payments

ER — Orders Service (MySQL)



ER — Payment Service (MySQL)

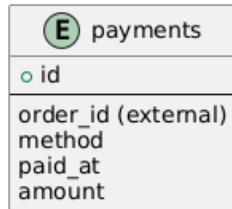
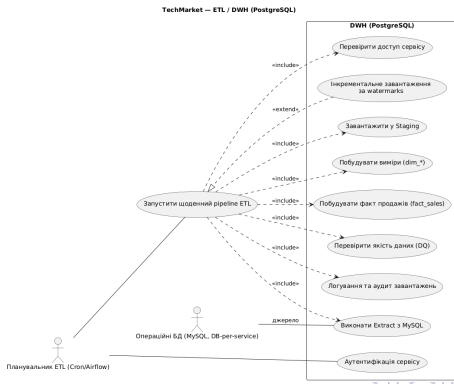


Рис.: Payments Service

Інструменти: Apache Airflow, Python (Pandas).

Етапи:

- 1 **Extract:** Вивантаження даних з MySQL (інкрементальне).
- 2 **Transform:** Очищення, денормалізація, агрегація.
- 3 **Load:** Завантаження в PostgreSQL (DWH).



Структура DWH (Схема "Зірка")

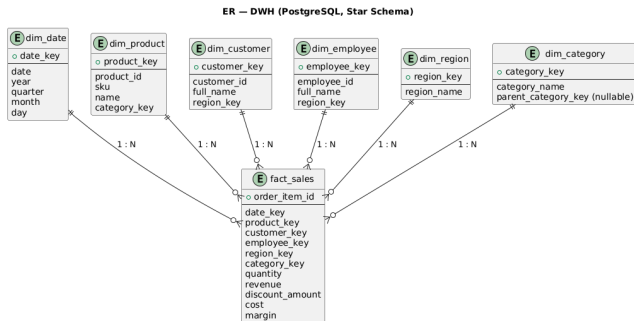


Рис.: Повна ER-діаграма сховища даних

Завдання: Розробка концептуальної та логічної моделі БД.

Результат:

- Створено ER-діаграми для 4 мікросервісів.
- Нормалізовано до 3НФ.
- Визначено зв'язки між сутностями.

Завдання: Розгортання OLTP БД у Docker.

Результат:

- Створено docker-compose конфігурацію.
- Ініціалізовано схеми для всіх сервісів.
- Згенеровано тестові дані (10K+ записів).

Завдання: Побудова ETL процесу.

Результат:

- Реалізовано інкрементальне завантаження.
- Автоматизовано через Airflow DAG.
- Трансформація у схему "Зірка".

Завдання: Оркестрація контейнерів.

Результат:

- Створено Kubernetes manifests.
- Налаштовано PersistentVolumes.
- Забезпечено автоматичний restart сервісів.

Завдання: Побудова аналітичних запитів та візуалізація.

Результат:

- Створено SQL запити для аналізу продажів.
- Інтегровано Metabase для візуалізації.
- *[Placeholder: Metabase Dashboard Screenshots]*

Проблема: Повільні запити при великих обсягах даних (1M+ рядків).

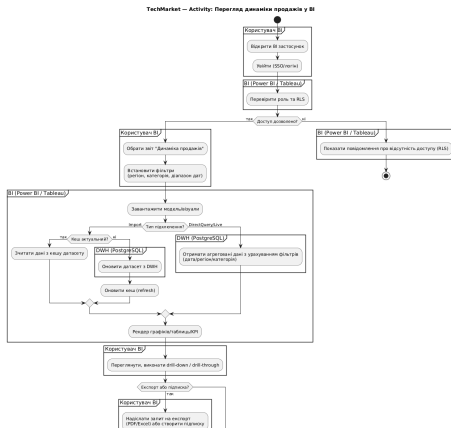
Застосовані методи:

- **Індексація:** B-Tree індекси для фільтрації та з'єднань.
- **Матеріалізовані представлення:** Кешування складних агрегацій.
- **VACUUM:** Очищення мертвих кортежів.

Результат: Прискорення запитів у **308x** (з 450ms до 1.46ms).

Аналітичні можливості:

- Аналіз трендів продажів.
- Сегментація клієнтів за активністю.
- Моніторинг інвентаря в реальному часі.



Що вдалося:

- Розробити реальну систему від проектування до оптимізації.
- Здобути досвід роботи з сучасними технологіями (Kubernetes, Airflow).
- Навчитись налагоджувати та оптимізувати складні запити.

Найбільші виклики:

- Налаштування Kubernetes — багато нюансів з volumes та networking.
- Оптимізація запитів на великих обсягах даних.

Висновок: Проект дав можливість об'єднати теоретичні знання з практикою та побудувати реальну інфраструктуру даних.

Дякую за увагу!