Архитектура базы данных

# 1. Общие положения

Каждый микросервис имеет собственную БД (подход database-per-service). Межсервисные связи реализуются через API, а не внешними ключами.

# 2. Структура БД по сервисам

## 2.1 catalog-db

- products — товары (id, sku, name, price, stock, is\_active).  
- categories — категории.  
- product\_categories — связь многие-ко-многим.  
- product\_images — изображения товара.

## 2.2 cart-db

- carts — корзины (id, user\_id, status).  
- cart\_items — позиции корзины (product\_id, qty, unit\_price, snapshot).

## 2.3 order-db

- orders — заказы (id, user\_id, total\_amount, status, addresses).  
- order\_items — позиции заказа (product\_id, name, price, qty).  
- order\_events — события заказа (логирование).

## 2.4 payment-db

- payments — платежи (id, order\_id, amount, status, provider).  
- payment\_transactions — транзакции (authorize, capture, refund).

## 2.5 auth-db (опционально)

- users — пользователи (id, email, password\_hash, role).

# 3. Основные правила

- Все PK — UUID.  
- Денежные поля — NUMERIC(12,2).  
- Валюты — CHAR(3), по умолчанию KZT.  
- Временные поля — TIMESTAMPTZ, хранение в UTC.  
- Миграции — Alembic.

# 4. Индексация

- products(sku), products(is\_active).  
- orders(user\_id, created\_at DESC).  
- order\_items(order\_id).  
- cart\_items(cart\_id).  
- payments(order\_id).

# 5. Тестовые данные

- 10+ товаров разных категорий.  
- Несколько корзин с товарами.  
- Заказы во всех статусах.  
- Платежи успешные и неуспешные.

# 6. Администрирование

- Ежедневные бэкапы (pg\_dump).  
- Хранение резервных копий ≥ 7 дней.  
- Мониторинг: количество заказов, успешные/неуспешные платежи.