

# Dokumentacja

## System Zarządzania Magazynem

Mykhailo Kleban, Yevhen Marchak

### Streszczenie

Celem projektu jest stworzenie systemu umożliwiającego ewidencję produktów, dostawców, magazynów, wysyłek i ruchów towarowych. System ma wspierać zarządzanie zapasami magazynowymi, kontrolę przyjmowania i wydawania towaru oraz generowanie raportów stanów magazynowych.

### Spis treści

<b>1 Wprowadzenie</b>	<b>2</b>
1.1 Opis systemu . . . . .	2
1.2 Zakres projektu . . . . .	2
<b>2 Wymagania systemowe</b>	<b>2</b>
2.1 Wymagania funkcjonalne . . . . .	2
2.2 Wymagania niefunkcjonalne . . . . .	3
<b>3 Projekt bazy danych</b>	<b>3</b>
3.1 Opis ogólny . . . . .	3
3.2 Tabele i relacje . . . . .	3
3.2.1 Tabela <code>categories</code> . . . . .	3
3.2.2 Tabela <code>suppliers</code> . . . . .	4
3.2.3 Tabela <code>clients</code> . . . . .	4
3.2.4 Tabela <code>employees</code> . . . . .	4
3.2.5 Tabela <code>locations</code> . . . . .	4
3.2.6 Tabela <code>products</code> . . . . .	5
3.2.7 Tabela <code>inventory</code> . . . . .	5
3.2.8 Tabela <code>receipts</code> . . . . .	5
3.2.9 Tabela <code>receipt_details</code> . . . . .	5
3.2.10 Tabela <code>shipments</code> . . . . .	6
3.2.11 Tabela <code>shipment_details</code> . . . . .	6
<b>4 Przepływ procesów magazynowych</b>	<b>6</b>
4.1 Proces przyjęcia dostawy (Receipt) . . . . .	6
4.2 Proces wydania / wysyłki (Shipment) . . . . .	6
4.3 Przesunięcia magazynowe / reorganizacja . . . . .	7

# 1 Wprowadzenie

## 1.1 Opis systemu

System Zarządzania Magazynem (SZM) to aplikacja mająca na celu usprawnienie procesów logistycznych wewnętrz magazynu. Umożliwia:

- rejestrowanie produktów i ich katalogowanie według kategorii,
- zarządzanie dostawcami,
- przyjmowanie dostaw i wystawianie przyjęć,
- kontrolę stanów magazynowych w poszczególnych lokalizacjach magazynu,
- planowanie i rejestrowanie wysyłek do klientów,
- monitorowanie pracowników zaangażowanych w operacje magazynowe,
- generowanie raportów i analizy przepływu towarów.

## 1.2 Zakres projektu

Projekt obejmuje:

1. projekt bazy danych, modelowanie relacji,
2. logikę obsługi magazynu (przyjęcia, wydania, przesunięcia),
3. interfejs użytkownika (frontend lub konsola / GUI),
4. raportowanie stanów magazynowych,

# 2 Wymagania systemowe

## 2.1 Wymagania funkcjonalne

- Rejestracja i edycja produktów (SKU, nazwa, waga, wymiary, opis).
- Przypisywanie produktów do kategorii oraz dostawców.
- Prowadzenie ewidencji stanów magazynowych — ilości, lokalizacje.
- Rejestrowanie przyjęć towarów od dostawców (receipts).
- Rejestrowanie wysyłek towarów do klientów (shipments).
- Śledzenie szczegółów przyjęć i wysyłek (ilość oczekiwana, ilość rzeczywista).
- Zarządzanie klientami i dostawcami (dane kontaktowe, dane podatkowe).
- Zarządzanie pracownikami (np. kto zatwierdza dokumenty magazynowe).
- Generowanie raportów stanu magazynu, przyjęć, wysyłek.

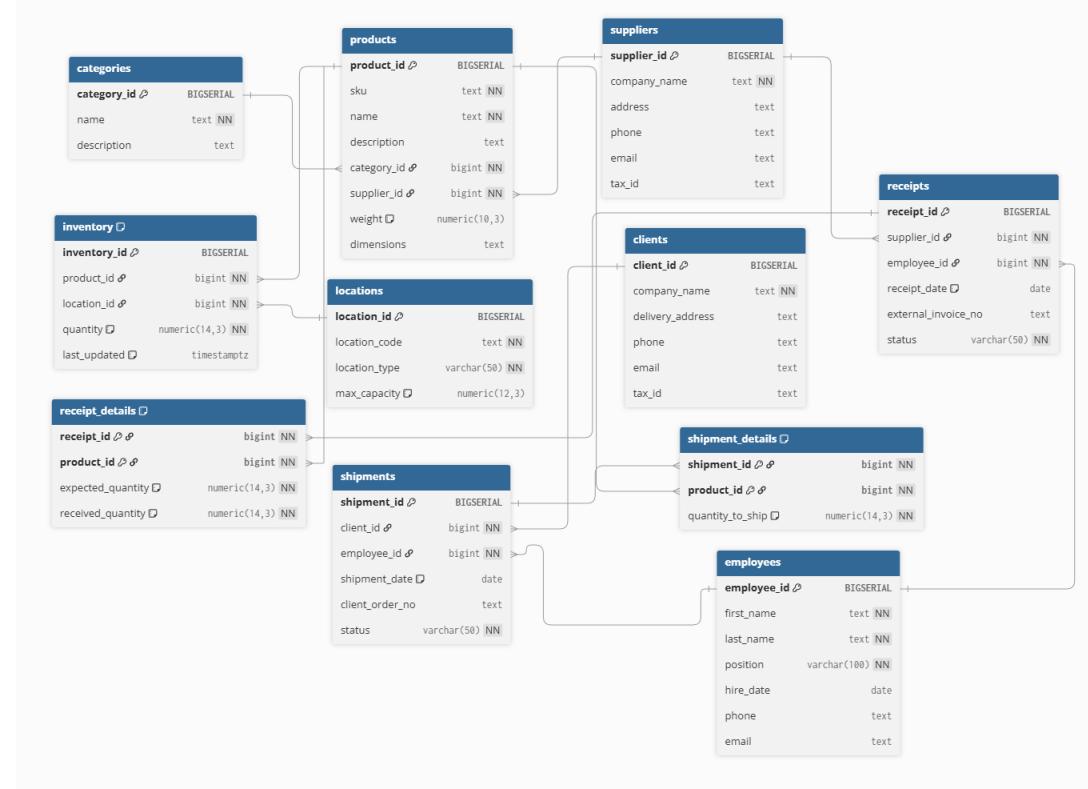
## 2.2 Wymagania niefunkcjonalne

- Wydajność: operacje ewidencji powinny być szybkie, nawet przy dużej liczbie rekordów.
- Skalowalność: system powinien wspierać rozbudowę magazynu (więcej lokalizacji, więcej produktów).
- Bezpieczeństwo: uwierzytelnianie użytkowników, ograniczanie dostępu do danych.
- Spójność danych: transakcyjność przy operacjach przyjęć i wydania.
- Możliwość tworzenia kopii zapasowych bazy danych.
- Łatwość integracji: API lub możliwość eksportu raportów.

## 3 Projekt bazy danych

### 3.1 Opis ogólny

System korzysta z relacyjnej bazy danych. Poniższy diagram przedstawia główne tabele systemu oraz relacje między nimi:



Rysunek 1: Diagram ERD – struktura bazy danych

### 3.2 Tabele i relacje

#### 3.2.1 Tabela categories

- `category_id` (BIGSERIAL, PK) — identyfikator kategorii

- **name** (text, NOT NULL) — nazwa kategorii
- **description** (text) — opis kategorii

### 3.2.2 Tabela suppliers

- **supplier\_id** (BIGSERIAL, PK) — identyfikator dostawcy
- **company\_name** (text, NOT NULL) — nazwa firmy dostawcy
- **address** (text) — adres
- **phone** (text) — telefon
- **email** (text) — e-mail
- **tax\_id** (text) — numer podatkowy

### 3.2.3 Tabela clients

- **client\_id** (BIGSERIAL, PK) — identyfikator klienta
- **company\_name** (text, NOT NULL) — nazwa firmy klienta
- **delivery\_address** (text) — adres dostawy
- **phone** (text) — telefon
- **email** (text) — e-mail
- **tax\_id** (text) — numer podatkowy

### 3.2.4 Tabela employees

- **employee\_id** (BIGSERIAL, PK) — identyfikator pracownika
- **first\_name** (text, NOT NULL) — imię
- **last\_name** (text, NOT NULL) — nazwisko
- **position** (varchar(100), NOT NULL) — stanowisko
- **hire\_date** (date) — data zatrudnienia
- **phone** (text) — telefon
- **email** (text) — e-mail

### 3.2.5 Tabela locations

- **location\_id** (BIGSERIAL, PK) — identyfikator lokalizacji w magazynie
- **location\_code** (text, NOT NULL) — kod lokalizacji
- **location\_type** (varchar(50), NOT NULL) — typ lokalizacji
- **max\_capacity** (numeric) — maksymalna pojemność lokalizacji

### 3.2.6 Tabela products

- **product\_id** (BIGSERIAL, PK) — identyfikator produktu
- **sku** (text, NOT NULL) — kod SKU produktu
- **name** (text, NOT NULL) — nazwa produktu
- **description** (text) — opis produktu
- **category\_id** (bigint, NOT NULL, FK → `categories.category_id`) — kategoria produktu
- **supplier\_id** (bigint, FK → `suppliers.supplier_id`) — dostawca produktu
- **weight** (numeric) — waga produktu
- **dimensions** (text) — wymiary produktu (np. „10x20x5 cm”)

### 3.2.7 Tabela inventory

- **inventory\_id** (BIGSERIAL, PK) — identyfikator rekordu stanów magazynowych
- **product\_id** (bigint, NOT NULL, FK → `products.product_id`) — produkt
- **location\_id** (bigint, NOT NULL, FK → `locations.location_id`) — lokalizacja magazynowa
- **quantity** (numeric, NOT NULL) — ilość produktu w danej lokalizacji
- **last\_updated** (timestampz) — data ostatniej aktualizacji stanu

### 3.2.8 Tabela receipts

- **receipt\_id** (BIGSERIAL, PK) — identyfikator przyjęcia
- **supplier\_id** (bigint, NOT NULL, FK → `suppliers.supplier_id`) — dostawca
- **employee\_id** (bigint, FK → `employees.employee_id`) — pracownik przyjmujący dostawę
- **receipt\_date** (date, NOT NULL) — data przyjęcia
- **external\_invoice\_no** (text) — numer faktury od dostawcy
- **status** (varchar(50), NOT NULL) — status przyjęcia (np. „oczekujące”, „zatwierdzone”)

### 3.2.9 Tabela receipt\_details

- **receipt\_id** (bigint, NOT NULL, FK → `receipts.receipt_id`)
- **product\_id** (bigint, NOT NULL, FK → `products.product_id`)
- **expected\_quantity** (numeric, NOT NULL) — oczekiwana ilość towaru
- **received\_quantity** (numeric) — faktycznie otrzymana ilość

### 3.2.10 Tabela shipments

- **shipment\_id** (BIGSERIAL, PK) — identyfikator wysyłki
- **client\_id** (bigint, NOT NULL, FK → `clients.client_id`) — klient, do którego wysyłka jest kierowana
- **employee\_id** (bigint, FK → `employees.employee_id`) — pracownik odpowiedzialny za wysyłkę
- **shipment\_date** (date, NOT NULL) — data wysyłki
- **client\_order\_no** (text) — numer zamówienia klienta
- **status** (varchar(50), NOT NULL) — status wysyłki (np. „w realizacji”, „wysłana”, „dostarczona”)

### 3.2.11 Tabela shipment\_details

- **shipment\_id** (bigint, NOT NULL, FK → `shipments.shipment_id`)
- **product\_id** (bigint, NOT NULL, FK → `products.product_id`)
- **quantity\_to\_ship** (numeric, NOT NULL) — ilość produktu do wysłania

## 4 Przepływ procesów magazynowych

### 4.1 Proces przyjęcia dostawy (Receipt)

1. Dostawca wysyła towar do magazynu.
2. Pracownik magazynu zakłada nowe „przyjęcie” w module `receipts`.
3. Dla każdego produktu w dostawie wprowadza się wiersze w `receipt_details` (ilość oczekiwana).
4. Po fizycznym rozładunku towaru, pracownik wprowadza faktyczne ilości otrzymane przez aktualizację `received_quantity`.
5. System aktualizuje `inventory`, doliczając ilość do odpowiednich lokalizacji magazynowych.
6. Stan przyjęcia może zmieniać się (np. „w trakcie”, „zatwierdzone”).

### 4.2 Proces wydania / wysyłki (Shipment)

1. Klient składa zamówienie.
2. Pracownik magazynu tworzy rekord w `shipments`, przypisując klienta i datę wysyłki.
3. Uzupełnia szczegóły wysyłki w `shipment_details` (które produkty i ile sztuk).

4. System sprawdza dostępny stan w magazynie (tabele `inventory`), rezerwuje lub oznacza, które lokalizacje będą użyte.
5. Pracownik pobiera towar, pakuje i wydaje.
6. Po dokonaniu wysyłki status zmienia się (np. „wysłana”, „dostarczona”).
7. System dekrementuje stany w `inventory` zgodnie z wysłanymi ilościami.

### **4.3 Przesunięcia magazynowe / reorganizacja**

Jeśli planujesz funkcjonalność przesunięć między lokalizacjami:

- Można dodać moduł „Transfery magazynowe” – rekordy przesunięć z lokalizacji A do lokalizacji B.
- Po zatwierdzeniu transferu system zmniejsza stan w lokalizacji źródłowej i zwiększa w docelowej.
- Wymaga to mechanizmu walidacji (maksymalne pojemności lokalizacji).