Błażej Pawluk, Dawid Dębkowski

**ShopDB**

**Spis treści:**

1. Opis celu aplikacji
2. Charakterystyka rodzajów użytkowników oraz wymagania funkcjonalne względem każdego z nich
3. Opis bazy danych
   1. opis wyodrębnionych encji oraz związków między nimi
   2. normalizacja bazy danych
   3. uzasadnienie przechowywania redundantnych danych
   4. opisy tabel, klucze, powiązania między tabelami
   5. diagramy
   6. triggery, procedury, widoki
4. Prawa dostępu użytkowników do poszczególnych komponentów bazy danych

**1. Opis celu aplikacji**

Aplikacja ShopDB jest symulacją internetowego sklepu odzieżowego. Pozwala na przeglądanie oraz kupowanie produktów dostępnych w sklepie. Ponadto jest ona również skierowana do pracowników sklepu – z poziomu aplikacji można dodawać nowe produkty, zarządzać stanem magazynu lub zamówieniami klientów.

**2. Charakterystyka rodzajów użytkowników oraz wymagania funkcjonalne względem każdego z nich.**

Przed wejściem do aplikacji należy się zalogować lub jeśli nie posiada się jeszcze konta – zarejestrować. Do każdego loginu przypisany jest jeden z typów użytkowników: klient, sprzedawca lub magazynier.

* **KLIENT** – klient sklepu. Rejestracja – stworzenie konta, z poziomu aplikacji tworzy domyślnie konto tego typu. Klient ma możliwość przeglądania katalogu produktów (przeszukiwania zgodnie z ustalonymi filtrami i opcjami sortowania), dodawania produktów o konkretnych rozmiarach do koszyka, a następnie do złożenia zamówienia. Takie zamówienie może on anulować lub opłacić, po czym zostanie ono do niego dostarczone (po skompletowaniu go przez magazyniera). Po dostarczeniu produktu klient może zgłosić zwrot zamówienia. Ponadto klient może dodawać do bazy swoje dane (bez których nie złoży zamówienia), przeglądać i edytować je, a także przeglądać swoje wcześniejsze zamówienia.
* **SPZEDAWCA** – pracownik sklepu. Po zalogowaniu się na swoje konto ma on możliwość przeglądania produktów, edytowania ich oraz dodawania nowych. Ponadto może dodawać nowe typy produktów oraz kolory do uwzględnienia w filtrach, zmieniać ceny produktów i ustawiać zniżki.
* **MAGAZYNIER** – pracownik magazynu. Po zalogowaniu się na swoje konto ma on możliwość przeglądania produktów w magazynie, może je dodawać, edytować ich ilość oraz usuwać z magazynu. Do jego obowiązków – funkcjonalności w aplikacji, należy również kompletowanie zamówień klientów oraz rozpatrywanie ich zwrotów.

**3. Opis bazy danych**

1. **Opis wyodrębnionych encji oraz związków między nimi:**

Użytkownicy – zawiera wszystkie dane o użytkownikach: loginy i hasła, typy użytkownika oraz jeżeli są klientem do dane do kontaktu, dostawy oraz wyboru użytkownika takie jak RODO, akceptacje regulaminu lub cookies.

Produkty – zawiera wszystkie informacje o produktach dostępnych w sklepie: nazwy, kategorie, typy, kolory, cenę, zdjęcia, rozmiary oraz stany magazynu o tych produktach

Zamówienia – zawiera wszystkie informacje o zamówieniach: kto je złożył, co zawierają, obecny status, faktury, wartość oraz logi o zmianach statusu zamówienia

Encja zamówienia „łączy” użytkowników z produktami – użytkownicy mają zamówienia złożone z produktów.

1. **Normalizacja bazy danych**

Wersja początkowa:

* **U** (użytkownicy) = (login, hasło, typ\_użytkownika, email, numer telefonu, typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, adres (ulica, numer\_domu, numer\_mieszkania, kod\_pocztowy, miasto), RODO, regulamin, cookies) – podkreślone atrybuty dotyczą tylko klientów, **FU** = {login → hasło, typ\_użytkownika; email → typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, adres, RODO, regulamin, cookies; numer\_telefonu → typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, adres, RODO, regulamin, cookies}
* **P** (produkty) = (nazwa, kategoria, typ, kolor, kod\_koloru, cena, zniżka, zdjęcia (lista), magazyn (rozmiar, stan\_magazynu, zarezerwowane)), **FP** = {nazwa, kategoria, typ, kolor → cena, zniżka, zdjęcia, magazyn, kolor → kod\_koloru, kod\_koloru → kolor}
* **Z** (zamówienia) = (id\_zamówienia, klient, pozycje (lista), status, faktura (id\_faktury, NIP, nazwa\_firmy, adres(ulica, numer\_domu, numer\_mieszkania, kod\_pocztowy, miasto)), wartość), **FZ** = {id\_zamówienia → klient, pozycje, status, faktura, wartość}
* **L** (logi zamówienia) = (id\_zamówienia, poprzedni\_status, nowy\_status, czas\_zmiany)

NF1 – każdy atrybut ma wartość atomową oraz nie istnieją powtarzające się krotki (istnieje klucz):

* **U** = (login, hasło, typ\_użytkownika, email, numer\_telefonu, typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, id\_adresu, RODO, regulamin, cookies), **FU** = {login → hasło, typ\_użytkownika; email → typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, id\_adresu, RODO, regulamin, cookies; numer\_telefonu → typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, id\_adresu, RODO, regulamin, cookies; imię, nazwisko}, **klucz minimalny**: (login, email)
* **A** (adresy) = (id\_adresu, ulica, numer\_domu, numer\_mieszkania, kod\_pocztowy, miasto), **FA** = {id\_adresu → ulica, numer\_domu, numer\_mieszkania, kod\_pocztowy, miasto}, **klucz**: (id\_adresu)
* **P** = (nazwa, kategoria, typ, kolor, kod\_koloru, cena, zniżka), **FP** = {nazwa, kategoria, typ, kolor → cena, zniżka, kolor → kod\_koloru, kod\_koloru → kolor }, **klucz**: (nazwa, kategoria, typ, kolor)
* **ZD** (zdjęcia) = (produkt, zdjęcie), **klucz**: (produkt, zdjęcie)
* **M** (magazyn) = (produkt, rozmiar, stan\_magazynu, zarezerwowane), **FM** = {produkt, rozmiar → stan\_magazynu, zarezerwowane}, **klucz**: (produkt, rozmiar)
* **Z** = (id\_zamówienia, klient, status, id\_faktury, wartość), **FZ** = {id\_zamówienia → klient, status, id\_faktury, wartość}, **klucz**: (id\_zamówienia)
* **PZ** (pozycje zamówienia) = (id\_zamówienia, produkt, rozmiar, ilość), **FPZ** = {id\_zamówienia, produkt, rozmiar → ilość}, **klucz**: (id\_zamówienia, produkt, rozmiar)
* **F** (faktura) = (id\_faktury, NIP, nazwa\_firmy, id\_adresu), **FF** = {id\_faktury → NIP, nazwa\_firmy, id\_adresu}, **klucz**: (id\_faktury)
* **L** = (id\_zamówienia, poprzedni\_status, nowy\_status, czas\_zmiany), **FL** = {id\_zamówienia, poprzedni\_status → nowy\_status, czas\_zmiany}, **klucz**: (id\_zamówienia, poprzedni\_status)

NF2 – jest w NF1 oraz wszystkie atrybuty niekluczowe są w pełni funkcyjnie zależne od klucza

* **U** = (login, hasło, typ\_użytkownika, email, numer\_telefonu, typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, id\_adresu, RODO, regulamin, cookies), **FU** = {login → hasło, typ\_użytkownika, email → numer\_telefonu, typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, id\_adresu, RODO, regulamin, cookies }, **klucz**: (login, email)
* **A** = (id\_adresu, ulica, numer\_domu, numer\_mieszkania, kod\_pocztowy, miasto), **FA** = {id\_adresu → ulica, numer\_domu, numer\_mieszkania, kod\_pocztowy, miasto}, **klucz**: (id\_adresu)
* **P** = (nazwa, kategoria, typ, kolor, cena, zniżka), **FP** = {nazwa, kategoria, typ, kolor → cena, zniżka }, **klucz**: (nazwa, kategoria, typ, kolor)
* **K** (kolory) = (kolor, kod\_koloru), **FK** = {kolor → kod\_koloru}
* **ZD** = (produkt, zdjęcie), **klucz**: (produkt, zdjęcie)
* **M** = (produkt, rozmiar, stan\_magazynu, zarezerwowane), **FM** = {produkt, rozmiar → stan\_magazynu, zarezerwowane}, **klucz**: (produkt, rozmiar)
* **Z** = (id\_zamówienia, klient, status, id\_faktury, wartość), **FZ** = {id\_zamówienia → klient, status, id\_faktury, wartość}, **klucz**: (id\_zamówienia)
* **PZ** = (id\_zamówienia, produkt, rozmiar, ilość), **FPZ** = {id\_zamówienia, produkt, rozmiar → ilość}, **klucz**: (id\_zamówienia, produkt, rozmiar)
* **F** = (id\_faktury, NIP, nazwa\_firmy, id\_adresu), **FF** = {id\_faktury → NIP, nazwa\_firmy, id\_adresu}, **klucz**: (id\_faktury)
* **L** = (id\_zamówienia, poprzedni\_status, nowy\_status, czas\_zmiany), **FL** = {id\_zamówienia, poprzedni\_status → nowy\_status, czas\_zmiany}, **klucz**: (id\_zamówienia, poprzedni\_status)

NF3 – jest w NF2 oraz wszystkie niekluczowe atrybuty zależą tylko od pełnego klucza minimalnego

* **U** = (login, hasło, typ\_użytkownika), **FU** = {login → hasło, typ\_użytkownika}, **klucz:** (login)
* **C** (klienci) = (email, numer\_telefonu, typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, id\_adresu, RODO, regulamin, cookies), **FC**  = {email → numer\_telefonu, typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, id\_adresu, RODO, regulamin, cookies, numer\_telefonu → email, typ\_klienta, imię, nazwisko, nazwa\_firmy, NIP, id\_adresu, RODO, regulamin, cookies}, **klucz**: (email, numer\_telefonu)
* **A** = (id\_adresu, ulica, numer\_domu, numer\_mieszkania, kod\_pocztowy, miasto), **FA** = {id\_adresu → ulica, numer\_domu, numer\_mieszkania, kod\_pocztowy, miasto}, **klucz**: (id\_adresu)
* **P** = (nazwa, kategoria, typ, kolor, cena, zniżka), **FP** = {nazwa, kategoria, typ, kolor → cena, zniżka }, **klucz**: (nazwa, kategoria, typ, kolor)
* **K** (kolory) = (kolor, kod\_koloru), **FK** = {kolor → kod\_koloru}
* **ZD** = (produkt, zdjęcie), **klucz**: (produkt, zdjęcie)
* **M** = (produkt, rozmiar, stan\_magazynu, zarezerwowane), **FM** = {produkt, rozmiar → stan\_magazynu, zarezerwowane}, **klucz**: (produkt, rozmiar)
* **Z** = (id\_zamówienia, klient, status, id\_faktury, wartość), **FZ** = {id\_zamówienia → klient, status, id\_faktury, wartość}, **klucz**: (id\_zamówienia)
* **PZ** = (id\_zamówienia, produkt, rozmiar, ilość), **FPZ** = {id\_zamówienia, produkt, rozmiar → ilość}, **klucz**: (id\_zamówienia, produkt, rozmiar)
* **F** = (id\_faktury, NIP, nazwa\_firmy, id\_adresu), **FF** = {id\_faktury → NIP, nazwa\_firmy, id\_adresu}, **klucz**: (id\_faktury)
* **L** = (id\_zamówienia, poprzedni\_status, nowy\_status, czas\_zmiany), **FL** = {id\_zamówienia, poprzedni\_status → nowy\_status, czas\_zmiany}, **klucz**: (id\_zamówienia, poprzedni\_status)

1. **Uzasadnienie przechowywania redundantnych danych**

Pomimo, że każda relacja zawiera klucz, do każdej jednej dodany został atrybut id, przykładowo **U** = (id\_użytkownika, login, hasło, typ\_użytkownika). Zostało to wprowadzone, aby przeszukiwanie danych było efektywniejsze (łatwiej sprawdzać pola typu int niż np. varchar(255)). Ponadto w relacji **C** został dodany atrybut id\_użytkownika, aby połączyć tabele.

Pomimo, że przy przeszukiwaniu faktur można bez problemu otrzymać dane klienta składającego zamówienie (id\_faktury w **Z**, klient w **Z**, który łączy się z **C**, które zawiera dane), to są one dodatkowo przechowywane w tej samej relacji. Zostało to zaprojektowane w taki sposób, aby przy zmianie danych klientów nadal znać dane, na które została wystawiona faktura w przeszłości.