

Projekt systemu sklepu muzycznego

Autorzy:

Kamil Król

Józef Piechaczek

Spis treści

Projekt systemu sklepu muzycznego	3
1. Ogólny opis systemu.....	3
2. Część wewnętrzna.....	3
2.1 Magazynier	3
2.2 Kierownik magazynu.....	3
2.3 Manager.....	3
2.4 Administrator	3
3. Część zewnętrzna.....	4
3.1 Klient	4
4. Baza danych.....	4
4.1 Tabele	4
4.2 Triggery i procedury	6
4.3 Diagram encji.....	7

Projekt systemu sklepu muzycznego

1. Ogólny opis systemu

Celem systemu jest dostarczenie możliwości zarządzania sklepem muzycznym. System będzie zaimplementowany w języku Java w połączeniu z relacyjną bazą danych SQL. System składać będzie się z dwóch głównych części:

- Część wewnętrzna - jej celem jest zapewnienie możliwości obsługi zamówień oraz zarządzania stanem magazynu.
- Część zewnętrzna – jej celem jest zapewnienie możliwości przeglądania dostępnych produktów i składania zamówień przez klienta.

Obie części będą aplikacjami okienkowym i będą korzystać ze wspólnej bazy danych.

2. Część wewnętrzna

Część wewnętrzna odpowiada za przetwarzanie zamówień złożonych w sklepie stacjonarnym oraz za zarządzanie stanami magazynowymi.

2.1 Magazynier

Opis wymagań funkcjonalnych

- Wyświetlanie istniejących zamówień w celu ich skompletowania.
- Zmiana statusu zamówienia na "przygotowano do wysyłki".

Wymagania funkcjonalne na poziomie bazy danych

- SELECT: orders, storage, products, albums, instruments, other_products
- UPDATE: orders

2.2 Kierownik magazynu

Może wykonywać wszystkie operacje możliwe dla magazyniera.

Opis wymagań funkcjonalnych

- Wprowadzanie nowych dostaw: zmiana stanów magazynowych

Wymagania funkcjonalne na poziomie bazy danych

- UPDATE: magazyn

2.3 Manager

Może wykonywać operacje możliwe dla kierownika magazynu.

Opis wymagań funkcjonalnych

- Dodawanie nowych oraz usuwanie/zmiana istniejących produktów.
- Anulowanie istniejącego zamówienia.
- Dodawanie nowych oraz usuwanie/zmiana istniejących opcji dostawy.

Wymagania funkcjonalne na poziomie bazy danych

- UPDATE: products, artists, albums, other, instruments, instrument_manufacturers, delivery
- INSERT: products, artists, albums, other, instruments, instrument_manufacturers, delivery
- DELETE: products, artists, albums, other, instruments, instrument_manufacturers, delivery

2.4 Administrator

Może wykonywać operacje możliwe dla menedżera.

Opis wymagań funkcjonalnych

- Dodawanie nowych oraz usuwanie/zmiana istniejących użytkowników
- Resetowanie haseł

Wymagania funkcjonalne na poziomie bazy danych

- UPDATE: users
- INSERT: users
- DELETE: users

3. Część zewnętrzna

Aplikacja dostarcza interfejs dla klienta i umożliwia mu przeglądanie produktów dostępnych w sklepie i składanie zamówień. Ta część systemu zakłada obecność tylko jednego użytkownika – klienta.

3.1 Klient

Ogólne wymagania i ograniczenia funkcjonalne

- Strona główna
- Wyświetlenie informacji o sklepie
- Przeglądanie produktów według kategorii
- Wyszukiwanie produktów według nazwy
- Dostarczenie informacji o możliwości zakupu danego produktu
- Kompletowanie zamówienia
- Wprowadzanie danych do zamówienia (imię, nazwisko, telefon, adres)
- Przeglądanie sposobów dostawy
- Podgląd i możliwość edycji skompletowanego zamówienia (przed jego złożeniem)
- Składanie zamówienia

Wymagania i ograniczenia funkcjonalne na poziomie bazy danych

- Select na tabeli products, albums, artists, instruments, instrument_manufacturers, other_product (w ramach przeglądania produktu), delivery (w ramach składania zamówienia)
- Insert na tabeli orders (po zatwierdzeniu zamówienia)

4. Baza danych

4.1 Tabele

W bazie danych będzie istnieć 12 tabel. Domyślnie żadna z kolumn nie może przyjąć wartości null, chyba że napisano inaczej.

- **Tabela 'products'**
zawiera wszystkie produkty dostępne w sklepie, informację o ich kategorii oraz cenie. Kluczem głównym jest 'product_id'. Poza kluczem głównym posiada jeszcze kolumnę 'category' oraz kolumnę 'price' mówiące odpowiednio o kategorii produktu i o jego cenie. Typ danych w kolumnie 'category' będzie typem wyliczeniowym przyjmującym wartości ze zbioru {album, instrument, other}.
- **Tabela 'albums'**
będzie zawierać wszystkie albumy dostępne w sklepie i informacje o nich. Kolumna 'album_id' będzie kluczem obcym do tabeli 'products' i kluczem głównym dla 'albums', kolumna 'artist_id' będzie kluczem obcym do tabeli 'artists'. Pozostałe kolumny to 'duration' zawierająca informację o długości całego albumu w sekundach, 'release_year' mówiąca o dacie premiery albumu, 'title' o tytule, 'songs_count' o liczbie piosenek, 'img_link' posiadająca link do obrazka, 'description' zawierająca informację o piosenkach na albumie. Kolumna 'img_link' może przyjmować wartość null

- **Tabela 'artists'**
kluczem głównym jest kolumna 'artist_id', kolumna 'genre' zawiera informacje o gatunku jaki tworzy artysta. Ponadto tabela zawiera kolumnę 'name' mówiącą o nazwie artysty. Tabela zawiera informacje o artystach, których albumy są w bazie.
- **Tabela 'instruments'**
kluczem głównym jest kolumna 'instrument_id', która jest też kluczem obcym do tabeli 'products', kolumna 'producer_id' jest kluczem obcym do tabeli 'instrument_manufactures'. Kolejne kolumny to 'type' mówiąca o typie instrumentu i kolumna 'name'. Tabela zawiera informacje o instrumentach dostępnych w sklepie.
- **Tabela 'instrument_manufacturers'**
kluczem głównym jest 'manufacturer_id'. Kolejna kolumna to 'name' mówiąca o nazwie instrumentów. Tabela zawiera informację o producentach instrumentów muzycznych.
- **Tabela 'other'**
jej kluczem głównym będzie kolumna 'product_id', będąca też kluczem obcym do tabeli 'product'. Pozostałe kolumny to 'producer' mówiąca o producencie danego produktu, 'name' mówiąca o nazwie i 'type' mówiąca o typie. Celem tej tabeli jest zapewnienie możliwości dodania produktów innego typu niż album czy instrument.
- **Tabela 'storage'**
zawiera informację o stanach magazynowych produktów. Pierwszą kolumną jest 'product_id', która stanowi klucz obcy do tabeli 'products', będąc jednocześnie kluczem głównym. Kolejna kolumna 'products_avaiable' określa ilość dostępnych produktów.
- **Tabela 'delivery'**
zawiera informację o dostępnych opcjach dostawy. Kluczem głównym jest kolumna 'delivery_id' określająca ID danej opcji. Kolumna 'name' zawiera nazwę danej dostawy. Kolumna 'price' przechowuje informacje o cenie danej opcji dostawy. Ostatnia kolumna 'delivery_time' zawiera czas realizacji dostawy podany w liczbie dni.
- **Tabela 'orders'**
zawiera informację o danym zamówieniu. Kluczem głównym jest 'order_id'. Kolejną kolumną jest 'current_status' będącą typem wyliczeniowym przyjmującym wartości z zakresu { in_progress, prepared_for_sending, sent, delivered, return, cancelled }. Informacja o wybranym dowodzie zakupu znajduje się w kolumnie 'transaction_document', która jest typem wyliczeniowym przyjmującym wartości z zakresu {invoice, receipt}. Kolumna 'delivery_id' jest kluczem obcym do tabeli 'delivery'. Kolejne kolumny zawierają inne, wybrane parametry zamówienia: 'payment' będące typem wyliczeniowym przyjmującym wartości z zakresu {on_delivery, bank_transfer}, 'customer_name', 'customer_address', 'phone_number'.
- **Tabela 'orders_products'**
tabela składa się z dwóch kolumn: 'order_id' będąca kluczem obcym do tabeli 'orders' oraz 'product_id' będąca kluczem obcym dla tabeli 'products'. Klucz główny składa się z obu kolumn. Tabela jest tabelą łączącą tabelę 'orders' z tabelą 'products'.

- **Tabela 'status_logs'**

tabela przechowuje logi dotyczące zmian statusu dokonanych w tabeli 'orders'. Pierwsza kolumna 'order_id' jest kluczem obcym do tabeli 'orders'. Kolejna kolumna 'user_id' jest kluczem obcym do tabeli 'users' i przechowuje informację o użytkowniku, który dokonał zmian. Kolumny 'status_old' oraz 'status_new' zawierają informację o zmianie statusu (mogą przyjmować wartość NULL) i są typem wyliczeniowym przyjmującym wartości z zakresu { in_progress, prepared_for_sending, sent, delivered, return, cancelled }. Kolumna 'date_time' zawiera informację o czasie dokonania zmiany. Klucz główny składa się z kolumn 'order_id' oraz 'user_id'.

- **Tabela 'users'**

tabela przechowuje dane użytkowników. Użytkownikiem jest osoba mająca dostęp do systemu wewnętrznego. Kluczem głównym tabeli jest kolumna 'user_id'. Jej wartość jest loginem danego użytkownika. Kolejna kolumna to 'password' - przechowuje informację o hasłach klienta w odpowiednio zabezpieczonej formie. Ostatnia kolumna 'access_level' przechowuje informację o roli użytkownika w firmie. Jej wartość należy do zbioru {storekeeper, storage manager, manager, administrator}.

4.2 Triggery i procedury

- ON INSERT i UPDATE na tabeli 'orders' - tworzenie logów zmiany statusu zamówienia, ewentualna zmiana tabeli 'orders-products'
- ON DELETE na tabeli 'albums' – usuwa artystę jeśli nie ma w bazie żadnego jego albumu oraz usuwa album z tabeli 'products', możliwe jeżeli dany album (będąc jednocześnie produktem) nie występuje w tabeli 'orders' ani 'orders-products'*
- ON DELETE na tabeli 'instruments' – usuwa producenta instrumentu jeżeli nie ma w bazie żadnego jego instrumentu oraz usuwa instrument z tabeli 'products', możliwe jeżeli dany instrument (będąc jednocześnie produktem) nie występuje w tabeli 'orders' ani 'orders-products'*
- ON DELETE na tabeli 'other' – sprawdza czy dany produkt nie występuje w tabeli 'orders' ani 'orders-products', jeżeli występuje to usunięcie jest niemożliwe*
- ON DELETE na tabeli 'orders' – usunięcie odpowiednich rekordów w tabeli 'orders_products', jeżeli dane zamówienie istnieje w tabeli 'status_logs' to usunięcie jest niemożliwe*
- ON DELETE na tabeli 'delivery' – sprawdza czy dana opcja dostawy nie występuje w tabeli 'orders', jeżeli występuje to usunięcie jest niemożliwe*
- ON DELETE na tabeli 'users' – sprawdza czy dany użytkownik nie występuje w tabeli 'status_logs', jeżeli występuje to usunięcie jest niemożliwe*
- ON DELETE na tabeli 'products' – najpierw usuwa album/instrument/inny produkt z odpowiedniej tabeli, ponadto usuwa produkt z magazynu, w przypadku gdy produkt jest w jakimś zamówieniu usunięcie jest niemożliwe, w tym celu będzie stworzona procedura 'usuwania historii'
- ON UPDATE na tabeli 'products' – umożliwia zmianę jedynie ceny produktu, w przypadku próby zmiany kategorii produktu pojawia się błąd
- Procedura 'clear_history' – usuwa wszystkie anulowane i zakończone zamówienia oraz wszystkie logi, pozwala ona na usuwanie sposobów dostawy, produktów w tym albumów, instrumentów oraz innych produktów. Przewidziana jest funkcja, która będzie umożliwiała wyeksportowanie danych historycznych do pliku.
- ON UPDATE na 'albums', 'instruments' – jeżeli artysta lub producent instrumentu zostanie zmieniony to sprawdza czy należy go usunąć, jeżeli tak to usuwa go

- Procedura dodawania do tabeli 'albums' – najpierw tworzy produkt później album, jeżeli potrzebuje stworzyć dodatkowo artystę to tworzy go**
- Procedura dodawania do tabeli 'instuments' – najpierw tworzy produkt później instuement, jeżeli potrzebuje stworzyć dodatkowo producenta instrumentów to tworzy go**
- Procedura dodawania do tabeli 'other' – najpierw tworzy produkt, a później dopiero rekord w tabeli 'other'**

*na ominięcie tej blokady pozwala opcja usuwania historii

**jednym z założeń tych procedur jest obecność ceny jako argumentu i danych o artyście/producencie instrumentów, które są używane bądź nie w zależności od potrzeby.

Ponadto istnieje założenie o tym, że żaden z użytkowników nie będzie bezpośrednio zmieniać rekordów w tabelach 'artists', 'indstrument_manufacturers', 'orders_products'. Login użytkownika z tabeli 'users' będzie niezmienny.

4.3 Diagram encji

