

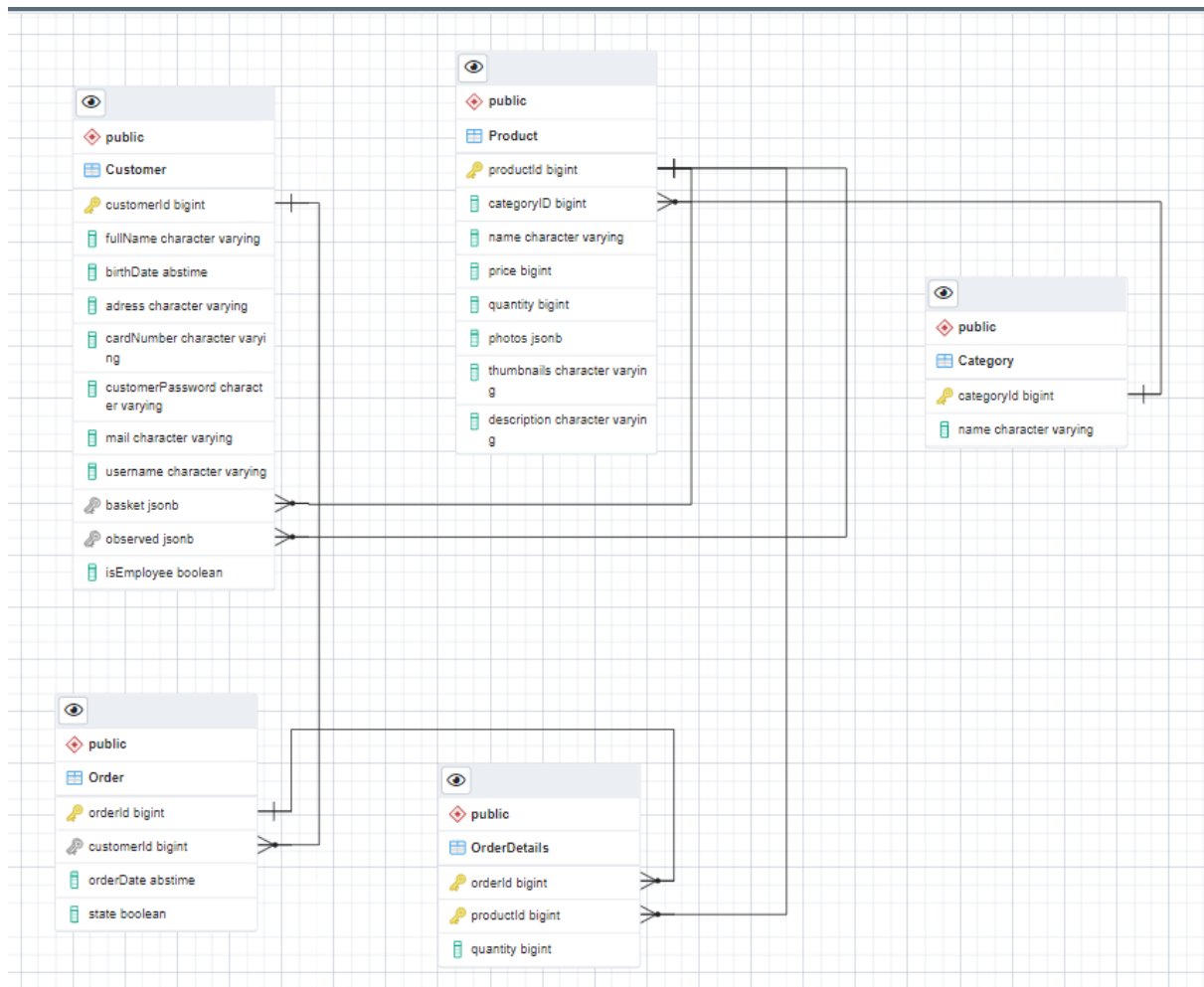
Bazy Danych - Projekt

Jakub Libera

Michał Siedlanowski

Robert Kubok

Diagram Bazy Danych



Opis Tabel:

Tabela Product – tabela przechowująca informację o produktach znajdujących się w sklepie

Klucz główny:

productId(bigint) – numer indywidualny produktu

Pozostałe:

categoryID(bigint) – numer indywidualny kategorii

name(char varying) – nazwa produktu

price(bigint) – cena produktu

quantity(bigint) – ilość dostępnych sztuk produktu

photos(jsonb) – zdjęcia produktu

thumbnails(character varying) – miniaturka głównego zdjęcia

description(character varying) – opis produktu

Tabela Category

Klucz główny:

categoryId(bigint) – numer indywidualny kategorii

Pozostałe:

name(character varying) - nazwa kategorii

Tabela Customer

Klucz główny:

customerId(bigint) - numer indywidualny klienta

Pozostałe:

fullName(character varying) - imię i nazwisko klienta

birthDate(date) - data urodzenia klienta

adress(character varying) - adres klienta

cardNumber(character varying) - numer karty klienta

customerPassword(character varying) - hasło klienta do konta

mail(character varying) - email klienta

username(character varying) - login klienta

basket(jsonb) - koszyk z produktami

observed(jsonb) - produkty obserwowane przez klienta

isEmployee(boolean) - czy jest pracownikiem

Tabela Order

Klucz główny:

orderId(bigint) - numer indywidualny zamówienia

Pozostałe:

customerId(bigint) - numer indywidualny klienta

orderDate(date) - data złożenia zamówienia

state(boolean) - stan realizacji zamówienia

Tabela OrderDetails

Klucz główny:

orderId(bigint) - numer indywidualny zamówienia

productId(bigint) - numer indywidualny produktu

quantity(bigint) - ilość zakupionych sztuk produktu

Opis Procedur:

clearbasket(

cid(bigint) - numer indywidualny klienta

) procedura czyści koszyk klienta o danym cid, wołana jest po kupnie przedmiotów z koszyka albo bezpośrednio przez klienta.

clearobserved(

cid(bigint) - numer indywidualny klienta

) procedura czyści listę obserwowanych produktów przez klienta o danym cid, wołana jest przez klienta.

insertcategory(

categoryname(text) - nazwa kategorii

) procedura dodaje nową kategorię, o ile nie istnieje. Wołana przez pracownika.

insertproduct(

categoryid(bigint),

productname(character varying),

price(bigint),

quantity(bigint),

thumbnails(character varying),

description(character varying)

) procedura dodaje nowy produkt do list. Wołana przez pracownika.

Opis Funkcji:

buyitemsfrombasket (

cid(bigint) - numer indywidualny klienta,

returns void

) funkcja wołana przez klienta, sprawdza możliwość kupna przedmiotów z koszyka klienta o danym cid. Jeśli któregoś z produktów kupić nie można zwraca error. Jeśli wszystkie produkty są dostępne, dodaje nowy zakup i odpowiednie produkty do historii zakupów oraz zmniejsza ilość dostępnych produktów w sklepie.

getuserbasketproducts(

cid(bigint) - numer indywidualny klienta,

returns jsonb

) wołana przez system przy zalogowaniu, zwraca dla danego klienta informacje o produktach w koszyku na podstawie **basket**(jsonb) z tabeli **Customer**.

getuserobservedproducts(

cid(bigint) - numer indywidualny klienta,

returns jsonb

) wołana przez system przy zalogowaniu, zwraca dla danego klienta informacje o produktach w liście obserwowanych produktów na podstawie **observed**(jsonb) z tabeli **Customer**.

customerhistory(

cid(bigint) - numer indywidualny klienta,

returns table

) wołana przez system przy zalogowaniu, zwraca dla danego klienta jego historie zakupów.

Serwer REST

Endpointy możemy podzielić na dwa rodzaje: wymagające autoryzacji poprzez przekazanie tokena użytkownika w nagłówku Authorization i niewymagające autoryzacji. Endpointy wymagające autoryzacji będą oznaczane przez dopisek AUTH. Wszystkie endpointy wymagające autoryzacji zwracają następujące statusy:

- OK - pomyślna autoryzacja
- UNAUTHORIZED - brak nagłówka Authorization lub brak użytkownika w bazie
- BAD REQUEST - niepoprawnie sformatowany nagłówek Authorization
- UNPROCESSABLE ENTITY - nieprzetwarzalny token

Endpointy

POST /api/register

Endpoint służy do rejestrowania użytkowników w bazie danych.

Przyjmuje

- Dane użytkownika: e-mail, imię i nazwisko, hasło, datę urodzenia, nazwę użytkownika w formacie JSON.

Zwraca

- BAD REQUEST - jeżeli brakuje danych lub są niepoprawne
- NOT ACCEPTABLE - jeżeli użytkownik już istnieje lub nie został przyjęty przez bazę danych
- OK - zwraca token nowo utworzonego użytkownika

POST /api/authorize

Endpoint służący do autoryzacji użytkownika, oczekuje w ciele zawartości typu JSON z polami email i password.

Zwraca

- OK - pomyślna autoryzacja, wraz z tokenem użytkownika
- BAD REQUEST - brak oczekiwanych danych
- NOT ACCEPTABLE - niepoprawne dane (użytkownika nie ma w bazie)

GET /api/user AUTH

Zwraca

- OK - dane użytkownika (JSON)

GET /api/user/basket AUTH

Zwraca

- OK - kolekcja produktów w koszyku użytkownika (JSON)

GET /api/user/observed AUTH

Zwraca

- OK - kolekcja produktów obserwowane przez użytkownika

GET /api/user/history AUTH

Zwraca

- OK - kolekcja produktów zakupionych kiedyś przez użytkownika (JSON)

PUT /api/user/basket AUTH

Endpoint służący do aktualizacji koszyka użytkownika.

Przyjmuje

- Nowy koszyk użytkownika: tablica obiektów z polami id reprezentujące produkty użytkownika np. [{ "id": "5" }]

Zwraca

- OK (text)

PUT /api/user/observed AUTH

Endpoint służący do aktualizacji obserwowanych produktów użytkownika.

Przyjmuje

- Nowa lista obserwowanych produktów: analogicznie do koszyka

Zwraca

- OK (text)

PUT /api/user/buy/basket AUTH

Endpoint służący do wykonania transakcji zakupu produktów znajdujących się aktualnie w koszyku użytkownika.

Zwraca

- OK (text)

GET /api/product

Endpoint służący do pobierania produktów z bazy. Obsługuje następujące parametry kwerendy:

- `limit` - maksymalna ilość wyświetlonych produktów (domyślnie 50)
- `page` - numer strony przy aktualnym limicie indeksowany od 0 (np. strona 1 przy limicie 50, zwróci produkty od 50 do 100)
- `categories` - lista nazw kategorii do których muszą należeć produkty np. `categories=laptopy,pralki` (domyślnie nie uwzględnia kategorii)

Zwraca

- OK - lista produktów zgodna z powyższymi parametrami kwerendy (JSON)
- NOT FOUND - nie znaleziono produktów dla powyższej kwerendy
- BAD REQUEST - wprowadzone w kwerendzie dane są niepoprawne

GET /api/product/{id}

Endpoint służący do pobierania danych dotyczących produktu o podanym identyfikatorze id.

Zwraca

- OK - dane produktu (JSON)
- NOT FOUND - brak produktu o podanym id

GET /api/category

Endpoint służący do pobierania kolekcji z nazwami kategorii.

Zwraca

- OK - lista nazw kategorii (JSON)

GET /api/category/{id}

Endpoint służący do pobierania nazwy kategorii o podanym id.

Zwraca

- OK - nazwa kategorii (JSON)
- NOT FOUND - nie odnaleziono kategorii o podanym id.

POST /api/product AUTH

Endpoint służący do dodawania produktów.

Przyjmuje

- Dane produktu w formacie JSON

Zwraca

- NOT FOUND - jeżeli podana kategoria produktu nie znajduje się w bazie
- CREATED - jeżeli produkt został pomyślnie utworzony (text)

POST /api/category AUTH

Endpoint służący do dodawania kategorii.

Przyjmuje

- Dane nowej kategorii

Zwraca

- OK - jeżeli kategoria została dodana

PUT /api/product/{id} AUTH

Endpoint służy do aktualizowania informacji o produktach.

Przyjmuje

- Nowe dane produktu

Zwraca

- NOT FOUND - jeżeli produkt o podanym id nie znajduje się w bazie
- OK (text)

GET /api/category/{id}

Endpoint służy do pobierania informacji o kategorii o podanym id.

Zwraca

- NOT FOUND - jeżeli kategoria o podanym id nie znajduje się w bazie
- OK - dane dotyczące kategorii

DELETE /api/product/{id} AUTH

Endpoint służy do usuwania produktu o podanym identyfikatorze id.

Zwraca

- FORBIDDEN - jeżeli użytkownik nie jest pracownikiem
- NOT FOUND - jeżeli produktu o podanym id nie było w bazie danych
- OK - jeżeli produkt zostanie usunięty