**Projekt z baz danych**

**Temat projektu:** *Aplikacja z interfejsem graficznym oraz obsługą bazy danych napisaną w języku PL/SQL*

**Tytuł projektu:** *Sklep komputerowy*

**Opiekun projektu:** *dr inż. Piotr Grochowalski*

**Autor projektu:** *Jakub Kuśnierz, Informatyka rok II*

**Data wykonania:** *semestr letni roku akademickiego 2016/2017*

**Użyte technologie/programy:**

- Baza danych została wykonana w Oracle Database z wykorzystaniem języka pl/sql

- Interfejs graficzny wykonany został w języku java

- Do pisania interfejsu wykorzystane zostały następujące technologie:

* Spring w wersji 4.2.5
* Spring Security w wersji 4.0.4
* Hibernate w wersji 5.1.0

- Do zarządzania projektem używany jest Maven w wersji 4.0

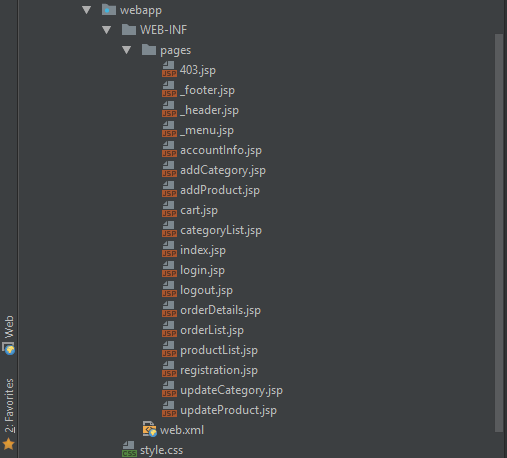
- Strona www wykonana została w technologii JSP (JavaServlet Pages)

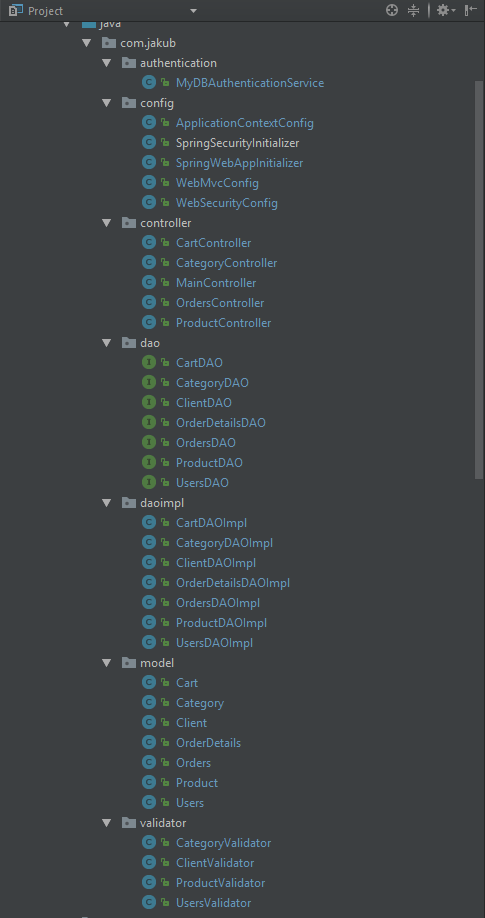
- Do uruchamiania aplikacji wykorzystany został Apache Tomcat w wersji 2.2

- Projekt napisany został w programie IntelliJ IDEA

Przy tworzeniu aplikacji wykorzystany został wzorzec projektowy **Model-View-Controller** który służy do organizowania struktur aplikacji posiadających graficzny interfejs użytkownika. Użyty został również komponent **Data Access Object** wykorzystywany jest do komunikacji między aplikacją a źródłem danych (np.. bazą danych). DAO stosowany w modelu MVC oddziela dostęp do danych od logiki warstwy prezentacji.

Aplikacja w języku java zawiera 29 klas, 7 interfejsów, 18 stron JSP oraz kilka plików konfiguracyjnych. Budowa aplikacji pokazana została na poniższych zdjęciach.

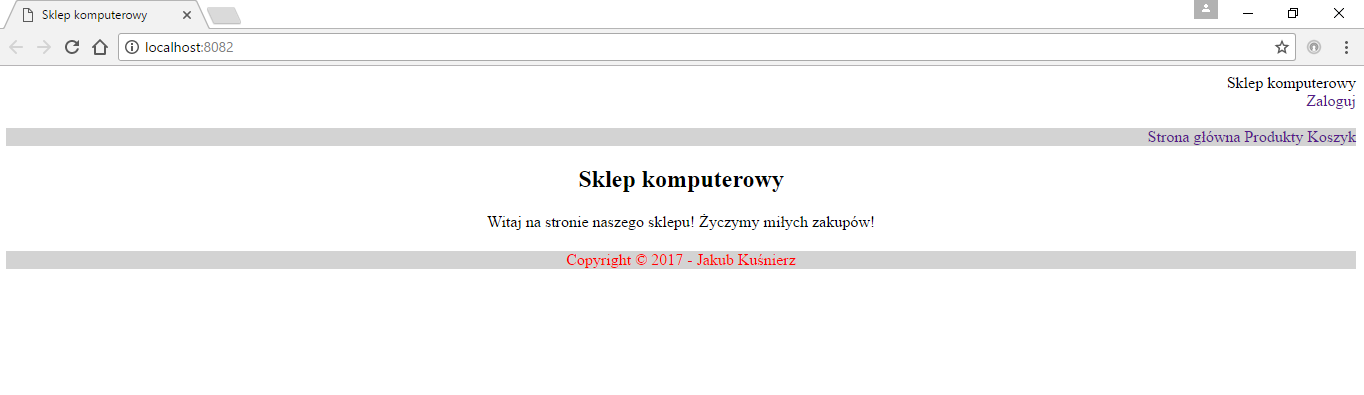




**Opis aplikacji**

Aplikacja webowa sklep komputerowy umożliwia dokonywanie zakupów przez internet oraz zarządzanie sklepem przez administratora. Jest to bardzo prosty model sklepu który po rozbudowaniu o możliwość dokonywania płatności przez kupujących mógłby zostać wykorzystany do prowadzenia własnego sklepu internetowego. Sklep jest przystosowany do handlu sprzętem komputerowym lecz do zmiany branży sprzedaży wystarczy wykonać edycję interfejsu, gdyż baza danych jest przystosowana do sprzedaży towarów innego rodzaju.

Po wejściu na stronę sklepu ukazuję się strona główna, z której mamy dostęp do zalogowania oraz przeglądania produktów sklepu.



**Opis poszczególnych funkcjonalności:**

1. **Logowanie**

W aplikacji istnieją dwa rodzaje kont użytkowników (klient i administrator). Po zalogowaniu na konto klienta dostajemy możliwość oglądania produktów, dodawania ich do koszyka oraz składania zamówień. Jako administratorzy dostajemy dostęp do dodatkowych opcji takich jak: tworzenie nowych produktów i kategorii, usuwanie oraz edytowanie produktów i kategorii, oglądanie listy wszystkich zamówień oraz podglądnie szczegółów zamówienia.

**Przykładowe dane logowania:**

Konto z prawami administratora

Login: *admin*

Hasło: *admin*

Konto klienta

Login: *test*

Hasło: *test*



Bezpośrednio ze strony logowania jest dostęp do przycisku Zarejestruj który przekierowuje nas na stronę wyświetlającą formularz rejestracji nowych klientów. Po wprowadzeniu w nim poprawnych danych (mechanizm walidacji danych sprawdza ich poprawność i wyświetla odpowiedni komunikat w przypadku błędnego uzupełnienia) zostaje utworzone w bazie konto nowego klienta.



Po poprawnym zalogowaniu na konto przycisk logowania zastępują dwa inne. Pierwszy w którym jest podana nazwa zalogowanego użytkownika przenosi nas na podstronę wyświetlającą informację o użytkowniku, natomiast drugi służy do wylogowywania z konta.



1. **Tworzenie nowego produktu**

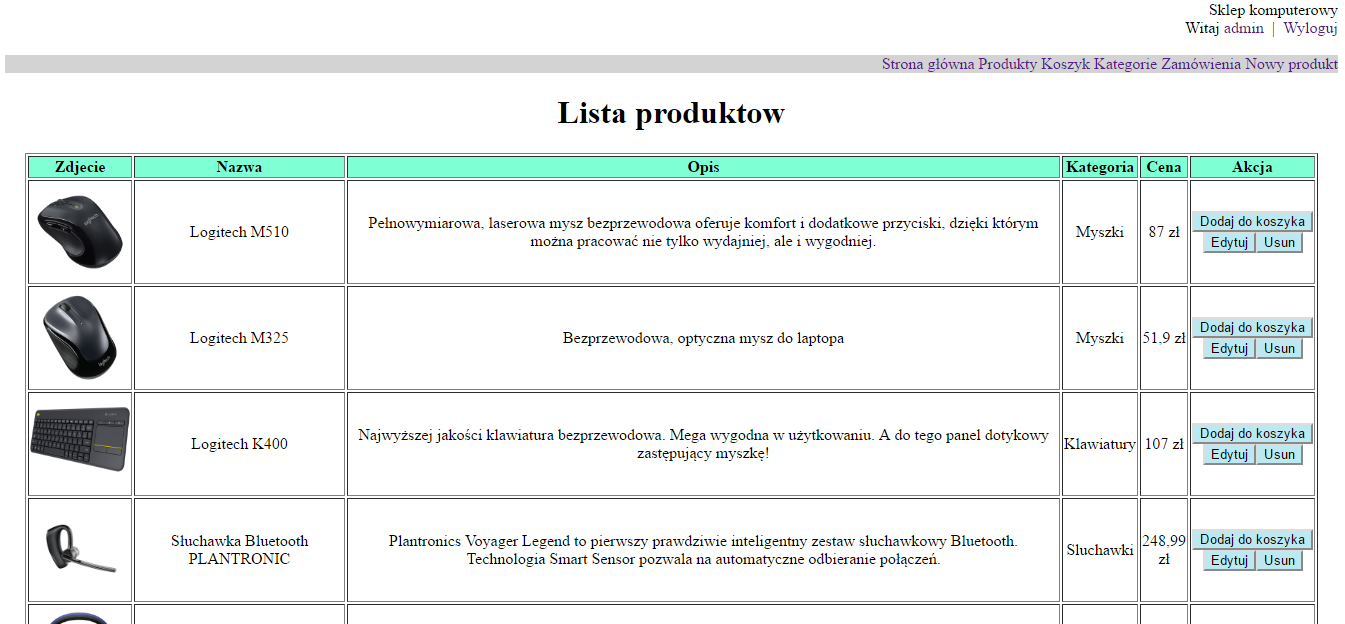
Jako administrator sklepu możemy tworzyć nowe produkty. Aby to zrobić wybieramy z menu nowy produkt i po otwarciu formularza uzupełniamy wymagane dane. Do tworzonego produktu możemy również dodać jego zdjęcie które zostanie wyświetlone na liście produktów.



1. **Wyświetlanie listy produktów**

Wybranie z menu przycisku produkty przenosi nas na stronę z listą produktów. Możemy na niej oglądać produkty które oferuje nasz sklep oraz dodawać je do naszego koszyka gdy jesteśmy zainteresowani zakupem.

Jako administratorzy otrzymujemy dostęp do dwóch kolejnych przycisków którymi są edytuj i usuń. Przycisk edycji przenosi nas na podstronę edycji produktu, a przycisk usuń usuwa produkt z bazy danych oraz z koszyków klientów jeżeli ktoś miał go dodanego.





1. **Kategorie**

Na podstronie kategorie naszym oczom ukazuję się tabela przedstawiająca listę wszystkich kategorii produktów w naszej bazie danych. Z podstrony kategorie możemy usuwać i edytować obecne kategorie.

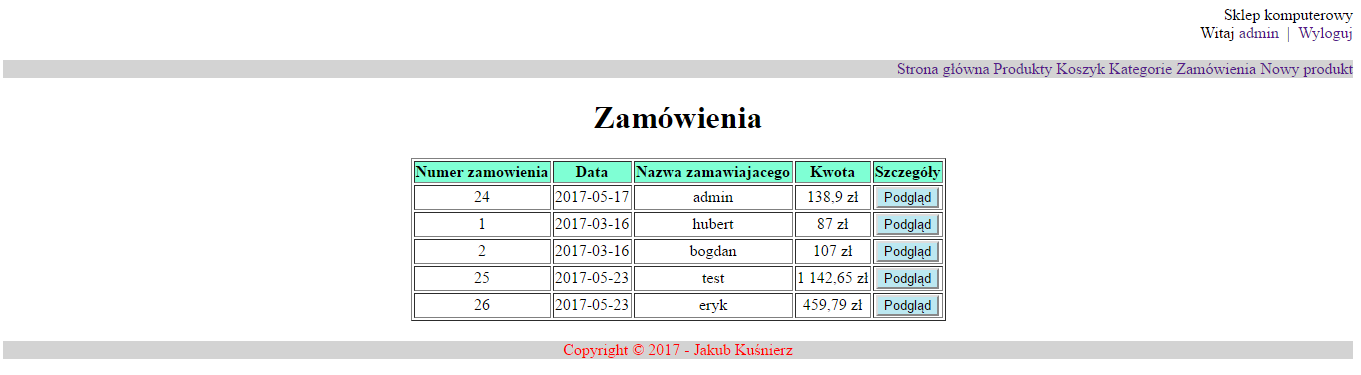


Z tej podstrony mamy bezpośredni dostęp do strony na której otrzymamy formularz do tworzenia nowej kategorii. Aby stworzyć kategorie musimy uzupełnić go poprawnymi danymi. Mechanizm walidacji sprawdzi czy wszystkie pola są poprawnie uzupełnione oraz czy podano minimalną ilość znaków w nazwie i opisie kategorii.



1. **Zamówienia**

Strona zamówienia dostępna jest tylko dla administratorów. Daje nam ona możliwość zobaczenia listy wszystkich zamówień złożonych przez klientów w naszym sklepie. W tabeli z listą otrzymujemy informacje o numerze zamówienia, osobie która go złożyła, dacie w której zostało złożone oraz o kwocie na jaką zostało złożone to zamówienie.



Przycisk podgląd przenosi nas do strony wyświetlającej szczegóły danego zamówienia. Informuje nas ona jakie produkty zostały kupione, ile sztuk każdego produktu zostało zamówione oraz jaka jest cena za sztukę każdego z produktów.



1. **Koszyk**

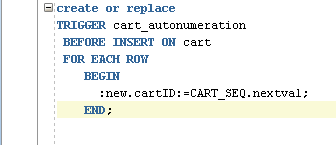
Strona koszyk przedstawia listę produktów które mamy w koszyku. Mimo iż jest to tylko jedna podstrona wykonuję się na niej najwięcej funkcji z bazy danych. Za pomocą odpowiednich przycisków możemy zwiększać oraz zmniejszać ilość sztuk produktów do kupna. Dzięki przyciskowi usuń możemy usuwać produkt z koszyka gdy zrezygnujemy z jego kupna. Kolejnymi funkcjami których nie możemy uruchomić za pomocą przycisków, ale które aktualizują się przy każdym przeładowaniu strony są obliczanie kwoty do zapłacenia za przedmioty w koszyku, naliczanie rabatu w zależności od wysokości kwoty zamówienia oraz obliczanie końcowej kwoty do zapłaty z uwzględnieniem otrzymanego rabatu. Ostatnią funkcją na tej podstronie jest możliwość złożenia zamówienia która dodaje nowe zamówienia do bazy danych, uzupełnia tabelę za szczegółami tego zamówienia oraz automatycznie opróżnia nasz koszyk abyśmy mogli dodawać do niego kolejne interesujące nas przedmioty i składać kolejne zamówienie.



**Wybrane fragmenty kodu w języku PL/SQL**

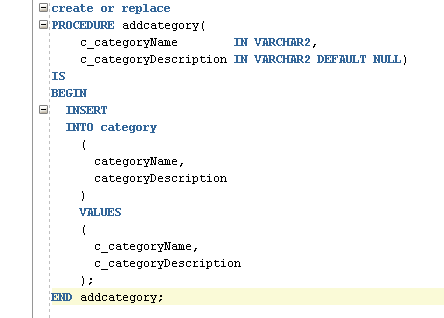
1. **Wyzwalacz autoinkrementcji id w tabeli koszyk**

Poniższy fragment kodu zapewnia autoinkrementacje klucza głównego w tabeli koszyk przy dodawaniu nowego przedmiotu do tabeli. Dzięki niemu przy tworzeniu procedur oraz funkcji dodających do tableli możemy pomijać w parametrach kluch główny, ponieważ zostanie on dodany automatycznie.

****

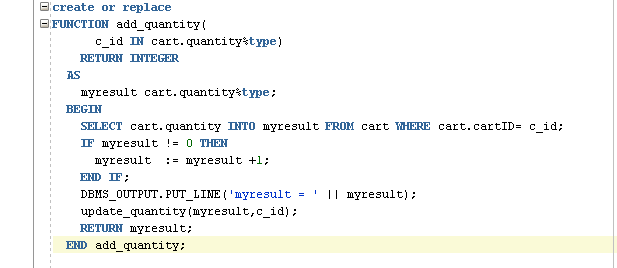
1. **Procedura dodawania do tabeli**

Kolejna prodcedura dodaje do tabeli kategorie nową kategorie. Przyjmuje ona dwa parametry wejściowe: nazwę oraz opis kategorii które następnie przypisuję do odpowiednich kolumn w tabeli. Klucz główny tabeli nie jest wymagany, ponieważ jest on tworzoni przez wyzwalacz, taki sam jak pokazany w powyższym przykładzie.



1. **Zwiększanie ilości sztuk produktu**

Funkcja ta zwiększa ilość sztuk produktu w koszyku, oraz zwraca aktualną ilość sztuk tego produktu. Do poprawnego działania funkcja potrzebuje 1 parametru wejściowego którym jest klucz główny koszyka w którym ma zostać zwiększona ilość. Ponadto funkcja wypisuje w konsoli bazy danych aktualną ilość sztuk. W ciele tej funkcji następuje wywołanie procedury która otrzymuje nową ilość sztuk przedmiotu i aktualizuje to w tabeli koszyk.



1. **Oblicznie rabatu**

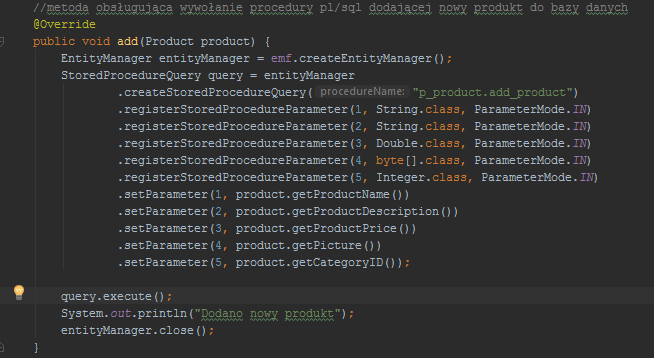
W tej funkcji na podstawie kwoty za całe zakupy w koszyku obliczany jest rabat dla klienta. Rabat uzależniony jest od wysokość kwoty do zapłaty i wynosi maksymalnie 10% całej kwoty. Do obliczania rabatu wykorzystana została instrukcja IF THEN. W funkcji została również funkcja agregująca SUM która oblicza sumę wartości dla wszystkich wierszy.

****

**Wybrane fragmenty kodu w języku java**

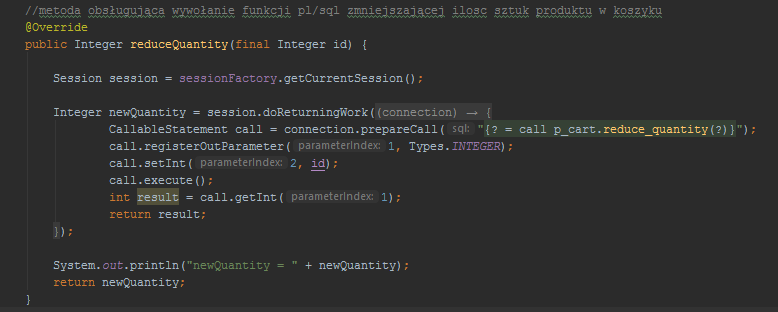
1. **Wywołanie procedury**

Do wywoływanie procedur pl/sql w projekcie najczęściej wykorzystywany został obiekt entity managera. Aby wywołać procedurę należy przypisać ją do obiektu klasy StoredProcedureQuery. W wywołaniu podajemy nazwe procedury. W naszym przypadku procedura znajduję się w pakiecie więc przy wywołaniu podajemy najpierw nazwę tego pakietu. Następnie rejestrujemy wszystkie parametry podane przy implementacji procedury określając ich typ oraz podając informację czy parametr jest wejściowy (IN) czy wyjściowy (OUT), parametr może być również połączeniem obu powyższych typów (INOUT). Po rejestracji parametrów musimy przypisać wartość parametrom wejściowym. Na koniec wykonujemy zapytanie do bazy oraz zamykamy obiekt entity menagera.

****

1. **Wywołanie funkcji**

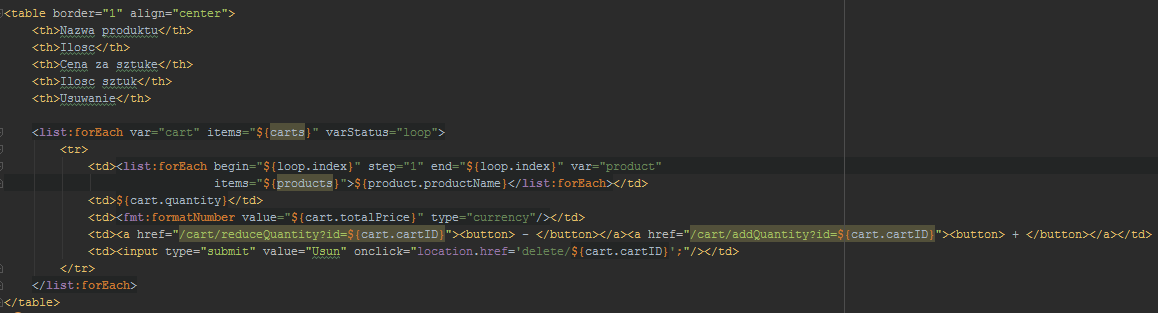
Metoda wywołująca funkcje pl/sql znacząco różni się od metody wywołującej procedure. Główną różnicą jest tutaj skorzystanie z obiektu session zamiast entity menagera. Wykorzystanie metody doReturningWork daję nam możliwość przekonwertowania sesii hibernate do połączenia JDBC. Daję nam to możliwość wywołania funkcji pl/sql a następnie zwrócenie wartości którą zwraca ta funkcja. Podobnie jak przy procedurze musimy zarejestrować wszystkie parametry oraz przypisać parametry wejściowe.

****

1. **Wyświetlenie tabeli w widoku JSP**

Strona JSP to dokument tekstowy który opisuje jak przetwarzać żadanie (request) aby otrzymać odpowiedź (response). Wyświetlanie elementów w stronach JSP odbywa się przy pomocy elementów języka HTML (lub XML) wzbogaconych o składniki pozwalające na dynamiczne generowanie zawartości strony. Do wyświetlenia tabeli używamy dyrektywy którą nazwaliśmy list (deklaracja na zdjęciu poniżej). Dyrektywa ta daje nam możliwość skorzystania z pętli forEach przy deklaracji której podajemy atrybut naszego modelu (w tym przypadku carts) po którym pętla będzie iterowała. Pętla wokonuję tyle iteracji ile ma obiektów w atrybucie, wyświetlając całą tabelę na ekran. W poniżym przykładzie wewnątrz pętli znaduję się kolejna która z innego atrybutu wyświetla nazwę przedmiotu znając wcześniej jego id. Na samym dole widzimy kod obsługujący przyciski w tabeli zwiększające i zmniejszające ilosć sztuk przedmiotu oraz usuwanie przedmiotu.





**Schemat bazy danych**

Baza danych wykorzystana przy projekcie składa się z 7 tabel. W tabeli koszyk wykorzystane zostały relacje wiele do wielu. W bazie przechowywane są następujące typy danych : Liczby całkowite oraz zmienno przecinkowe (number), ciągi znaków (varchar2), Data (date) oraz zdjęcia jako pliki binarne (format BLOB).

**Opis poszczgólnycyh tabeli**

1. **Klient (client)**

W tabeli tej przechowywane są szczegółowe dane użytkowników takie jak imie, nazwisko, email, adres zameldowania i kod pocztowy. Dane te przypisywane są do użytkowników z tabeli Users.

1. **Użytkownicy (users)**

Tabela ta przechowuje dane o użytkownikach służące do logowania (login i hasło), role użytkownika na stronie (klient lub administrator) oraz ID z tabeli klient dzięki któremu możemy sprawdzić szczegółowe

informacje o kliencie.

1. **Produkty (product)**

Kolejna z tabel przechowuje informacje o produktach sprzedawanych w naszym sklepie. Znajdziemy w niej informacje o nazwie, opisie oraz cenie produktu. Ponadto znajdujemy tu id które jest kluczem obcym z tabeli kategorie i mówi nam do której z nich należy dany produkt oraz zdjęcie produktu które użytkownik może dodać przy tworzeniu nowego produktu.

1. **Kategorie (category)**

Następna tabela jest tabelą w której znajdujemy informacje o kategoriach produktów w naszym sklepie. Jest ona najprostszą tabelą gdyż zawiera tylko trzy kolumny (klucz główny, nazwa kategorie oraz jej opis) i nie zawiera żadnego klucza obcego.

1. **Koszyk (cart)**

Tabela posiada relacjęwiele do wielu. Nie jest do zwykła tabela łączenia, ponieważ oprócz dwóch kluch obcych (id użytkownika oraz id produktu) zawiera także informację o ilości sztuk danego produktu oraz o cenie tego produktu. W momencie wyboru akcji dodania do koszyka zostaje utworzony w niej nowy rekord. Z kolei w momencie złożenia zamówienia przez użytkownika koszyk tego użytkownika automatycznie jest oprózniany.

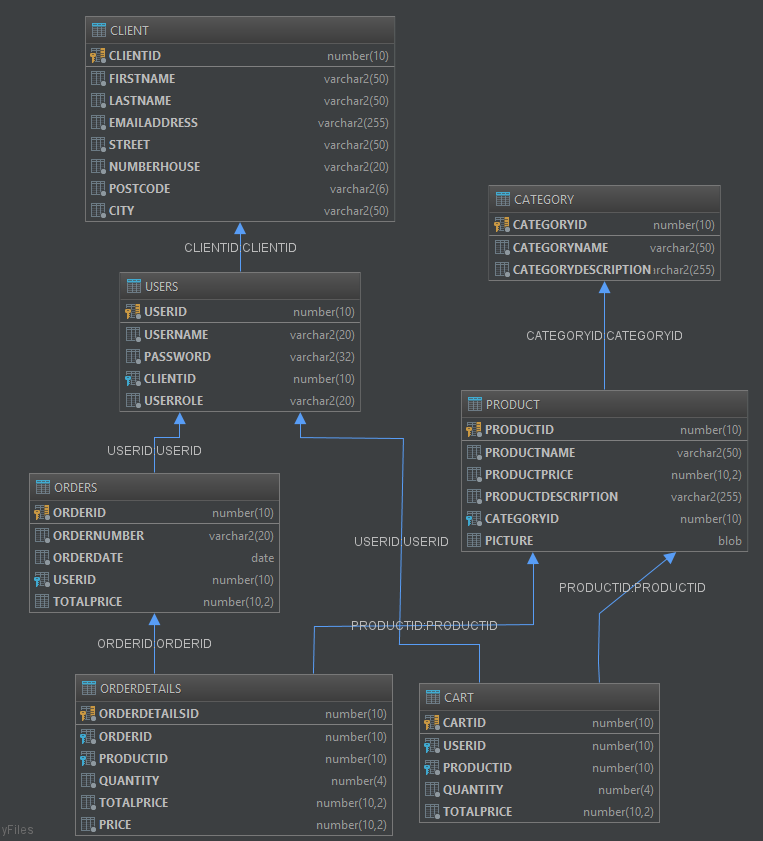
1. **Zamówienia (orders)**

Przedostatnia tabela zawiera informacje o złożonych zamówieniach. Możemy w niej zobaczyć całkowity koszt zamówienia, date złożenia, numer oraz użytkownika który dokonał tego zamówienia.

1. **Szczegóły zamówienia (orderDetails)**

Ostatnia z tabel bezpośrednio łączy się z tabelą zamówienia w relacji 1 do wielu. Możemy w niej sprawdzić szczegóły danego zamówienia, czyli jakie przedmioty były kupowane, w jakich ilościach oraz po jakiej cenie.

**Schemat encji bazy danych**



*Schemat encji wygenerowany został w programie IntellJ IDEA*