



**POLITECHNIKA LUBELSKA  
WYDZIAŁ ELEKTROTECHNIKI  
I INFORMATYKI**

**KIERUNEK STUDIÓW  
INFORMATYKA**

Przedmiot: Wprowadzenie do systemów baz danych

*Raport z wykonania projektu pt.*

**System wspomagania internetowych sklepów odzieżowych**

Autor:

*Jakub Biaduń*

*Paweł Bojarczuk*

*Frantsysk-Yan Boika*

*Dariusz Bochyński*

Lublin, 2023



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny



## **ROZDZIAŁ 1. OPIS WYBRANEGO OBSZARU RZECZYWISTOŚCI ORAZ WSKAZANIE PROBLEMU, KTÓRY ZOSTANIE ROZWIĄZANY PRZY POMOCY SYSTEMU INFORMATYCZNEGO WYKORZYSTUJĄCEGO PROJEKTOWANĄ BAZĘ DANYCH**

W ramach projektu prezentujemy internetowy sklep odzieżowy, który jest odpowiedzią na problem sprzedaży nieoryginalnej odzieży. Ten sklep funkcjonuje na terenie Unii Europejskiej. Sklep działa w wersji webowej i aplikacji mobilnej. Ułatwiamy klientom wyszukiwanie produktów poprzez różne filtry dostępne na sklepie takie jak kolor dominujący danej pary, rozmiar czy kategoria produktu. Kupujący może kontynuować swoje zakupy po zmianie platformy. Typy płatności obejmuje płatność kartą, Apple pay/Google pay, przelew oraz płatność przy odbiorze. Klienci otrzymują zamówiony produkt za pomocą dostawy kurierskiej bądź odbiór w paczkomacie. Na stronie będzie dostępna możliwość śledzenia przesyłki. Produkty w sklepie są objęte gwarancją, jest możliwy zwrot zamówionych produktów. Nasz system informatyczny ułatwia dostęp do zakupu markowej, oryginalnej odzieży.



## **ROZDZIAŁ 2. SFORMUŁOWANIE CELU BUDOWY SYSTEMU INFORMATYCZNEGO, WYKORZYSTUJĄCEGO PROJEKTOWANĄ BAZĘ DANYCH, ORAZ PREZENTACJA WYMAGAŃ FUNKCJONALNYCH STAWIANYCH SYSTEMOWI**

### **→ Konto:**

- ◆ Rejestracja - pozwala użytkownikom tworzenie konta w sklepie.
- ◆ Logowanie - umożliwia zalogowanie się na istniejące konto.
- ◆ Edycja konta - przypisanie preferowanej płatności, waluty, adresu wysyłki oraz podstawowe dane osobowe.
- ◆ Historia zamówień - dzięki niej klient ma łatwy wgląd do danych do faktury oraz może w prosty sposób złożyć reklamację bądź zwrócić przedmiot.
- ◆ Śledzenie przesyłki - klient ma możliwość śledzenia zamówionej przesyłki.
- ◆ Modyfikacja zabezpieczeń - zmiana hasła oraz całkowite usunięcie konta.
- ◆ Uprawnienia konta - są to role konta, które posiadają różny zakres uprawnień, np. użytkownik specjalny, taki jak administrator ma dodatkowe uprawnienia w serwisie takie jak modyfikacja listy produktów, zarządzanie sklepem internetowym.

### **→ Sklep:**

- ◆ Zakup produktów - użytkownik może dodać produkt do koszyka, po czym sfinalizować zamówienie.
- ◆ Koszyk - użytkownik może dodawać i usuwać produkty z koszyka.
- ◆ Filtrowanie produktów - wyszukiwanie produktów po rozmiarach, typie produktu oraz kolorach pozwala klientom na łatwiejsze poruszanie się po stronie.
- ◆ Dane odbiorcy i sposób płatności- pomimo przypisanych danych odbiorcy i preferowanej metody płatności do konta, klient podczas zakupu wciąż może je zmienić.
- ◆ Promocje - Mogą być zbiorowe np. o wybranej tematyce ale promocje mogą również obejmować pojedyncze produkty. Każda kategoria produktów może być objęta promocją. Promocje są dostępne w określonych ramach czasowych.

### **→ Produkty:**

- ◆ Informacje o produktach - takie jak np. kategoria, nazwa, cena, producent, kolor, rozmiar, zdjęcie i opis.
- ◆ Dostępność produktu - informacja czy dany produkt jest obecnie w sprzedaży.



- ◆ Opinie kupujących - jeżeli użytkownik zakupił dany produkt może napisać jego opinie.

**→ Reklamacja:**

- ◆ Możliwość reklamacji - klient w określonym czasie gwarancyjnym danego produktu, który został przez niego zamówiony, może dokonać reklamacji.
- ◆ Przebieg reklamacji - klient może odesłać zakupione produkty, które chce zareklamować, jeśli reklamacja zostanie rozpatrzona pozytywnie klient otrzyma nowy produkt lub zwrot środków (w przypadku opłaty przy odbiorze klient musi podać swój numer konta bankowego). Jeżeli reklamacja zostanie rozpatrzona negatywnie, klientowi zostaną odesłane reklamowane przez niego produkty.
- ◆ Status reklamacji - klient ma możliwość obserwowania aktualnego statusu procesu reklamacji.

**→ Zwrot:**

- ◆ Możliwość zwrotu - klient w określonym czasie po otrzymaniu zakupionego zamówienia, może dokonać zwrotu produktu lub produktów zawartych w zamówieniu bez podania przyczyny.
- ◆ Przebieg zwrotu - klient może odesłać zakupione produkty, które chce zwrócić, jeśli zwrot zostanie rozpatrzony pozytywnie klient otrzyma zwrot środków (w przypadku opłaty przy odbiorze klient musi podać swój numer konta bankowego). Jeżeli zwrot zostanie rozpatrzony negatywnie, klientowi zostaną odesłane zwracane przez niego produkty.
- ◆ Status zwrotu - klient ma możliwość obserwowania aktualnego statusu zwrotu.

### **ROZDZIAŁ 3. SZCZEGÓŁOWY OPIS PROCESÓW REALIZOWANYCH W WYBRANEJ RZECZYWISTOŚCI, ZALEŻNOŚCI MIĘDZY NIMI ORAZ OBIEKTÓW W NICH UCZESTNICZĄCYCH**

#### **→ Konto:**

- ◆ Rejestracja - Podanie danych konta (login, e-mail, hasło) oraz zaznaczenie zgody na akceptację regulaminu. Serwer sprawdza, czy konto na podanym e-mailu istnieje, a następnie wysyła link aktywacyjny na podany podczas rejestracji e-mail. Użytkownik musi kliknąć w wysłany link aby utworzyć konto.
  - Obiekty - Nowy użytkownik, lista użytkowników
- ◆ Logowanie - Użytkownik podaje swój login oraz hasło po czym serwer sprawdza ich poprawność. Jeśli dane są poprawne, użytkownik zostaje zalogowany. Użytkownik ma również opcje odzyskiwania hasła, aby z niej skorzystać musi kliknąć opcję “przypomnij hasło” i podać adres e-mail przypisany do istniejącego konta. Serwer sprawdza, czy konto na podanym e-mailu istnieje, a następnie wysyła link do zmiany hasła na wskazany adres e-mail.
  - Obiekty - Użytkownik, lista użytkowników
- ◆ Edycja konta - Po zalogowaniu się i przejściu w ustawienia konta, użytkownik ma dostęp do zmiany danych takich jak domyślny adres dostawy i preferowana metoda płatności, preferowana waluta. Aby zatwierdzić zmiany należy kliknąć przycisk “Potwierdź zmiany”. System sprawdza czy dane są poprawne po czym wprowadza je do systemu. Natomiast, jeśli wystąpią błędy użytkownik zostaje o tym poinformowany a zmiany nie zostaną zaakceptowane do systemu.
  - Obiekty - Użytkownik
- ◆ Historia zamówień - Aby móc zobaczyć historię zamówień, zalogowany użytkownik musi kliknąć zakładkę “Moje konto” a następnie “Historia zamówień”. W niej ma dostęp do informacji dotyczących numeru zamówienia, danych przesyłki i zarządzania przesyłką (zwrot, reklamacja).
  - Obiekty - Użytkownik, zamówienia
- ◆ Śledzenie przesyłki - Aby móc śledzić zamówienie, zalogowany użytkownik musi kliknąć zakładkę “Moje konto” a następnie “Historia zamówień”. Następnie klient musi wybrać zamówienie które chce śledzić. Następnie należy kliknąć “śledzenie listu przewozowego” który przekieruje klienta do zewnętrznej strony firmy kurierskiej realizującej to zamówienie.
  - Obiekty - Klient, zamówienie
- ◆ Modyfikacja zabezpieczeń - Po kliknięciu zakładki “Moje konto” a następnie “Bezpieczeństwo” klient może zmienić swoje hasło lub usunąć konto z serwisu.



**→ Sklep:**

- ◆ Zakupy - W celu dokonania zakupu, należy wybrać produkt który chcemy kupić i dodać go do koszyka. Następnie, przejść do finalizacji płatności gdzie wybieramy adres docelowy i metodę płatności.
  - Obiekty - Użytkownik, produkt, koszyk, zawartość koszyka
- ◆ Koszyk - Użytkownik może dodawać interesujące go produkty do koszyka, klikając w zakładce produktu przycisk "dodaj do koszyka". Użytkownik może przeglądać zawartość swojego aktualnego koszyka, może go edytować poprzez zmianę ilości sztuk produktu lub usunięcie produktu z koszyka. Zmieniając platformę sklepu koszyk użytkownika pozostaje taki sam.
  - Obiekty- Użytkownik, koszyk, zawartość koszyka
- ◆ Historia zamówień- Użytkownik może oglądać swoje zamówienia.
  - Obiekty- Użytkownik, zamówienia
- ◆ Filtrowanie produktów - W zakładce produkty jest możliwość filtrowania produktów według kryteriów zadanych przez użytkownika dotyczących np koloru, rozmiaru, marki. Aby zastosować filtrowanie należy użyć opcji "Filtruj" i wybrać kryteria które chcesz zastosować. Następnie użyć "Zastosuj"
  - Obiekty - Produkt, lista kategorii i podkategorii produktów
- ◆ Dane odbiorcy i sposób płatności - W procesie zakupu można wybrać metodę płatności takich jak karta kredytowa, Apple Pay/Google Pay, przelew oraz płatność przy odbiorze. Jeśli dane odbiorcy nie są zapisane na jego koncie, należy je wpisać w przeznaczonych do tego rubrykach.
  - Obiekty - Użytkownik, zamówienie, metoda płatności, waluta
- ◆ Promocje - Produkty objęte promocją znajdują się na stronie razem z innymi produktami które obecnie nie są przecenione. Jednak by znaleźć wyłącznie produkty które są obecnie na promocji trzeba wejść w zakładkę na stronie "Wyprzedaże". Zakup produktu w obniżonej cenie jest dostępny tylko podczas trwania promocji, w określonej ilości sztuk danego produktu.
  - Obiekty - produkt, promocje

**→ Produkty:**

- ◆ Informacje o produktach - dostarcza szczegółowe dane dotyczące produktu, takie jak nazwa produktu, model, cena, zdjęcie, opis, opinie.
  - Obiekty - model obuwia, kolor, marka, kategoria, podkategoria, promocja,
- ◆ Dostępność - Informuje o aktualnym stanie magazynowym produktu, czyli czy jest on dostępny do natychmiastowej wysyłki lub ile czasu zajmie jego dostarczenie w przypadku braku w magazynie.

- Obiekty - produkt, model, rozmiar
- ◆ Opinie - po zakupie i otrzymaniu produktu, klient ma możliwość wystawienia opinii zakupionemu produktowi.
  - Obiekty - zakupiony produkt, użytkownik

**→ Zwrot:**

- ◆ Możliwość zwrotu - klient w określonym czasie po otrzymaniu zakupionego zamówienia, może dokonać zwrotu produktu lub produktów zawartych w zamówieniu bez podania przyczyny.
  - Obiekty- użytkownik, zakupione produkty, zwrot, zamówienie
- ◆ Przebieg zwrotu - klient może odesłać zakupione produkty, które chce zwrócić. Otrzymuje on w wiadomości e-mail etykietę potrzebną do zwrotu przesyłki. Jeśli zwrot zostanie rozpatrzony pozytywnie klient otrzyma zwrot środków (w przypadku opłaty przy odbiorze klient musi podać swój numer konta bankowego). Jeżeli zwrot zostanie rozpatrzony negatywnie, klientowi zostaną odesłane zwracane przez niego produkty.
  - Obiekty- użytkownik, zakupione produkty, zwrot, metody płatności
- ◆ Status zwrotu - klient ma możliwość obserwowania aktualnego statusu zwrotu, po wybraniu zakładki “status” w menu zwrotu. Ma on wtedy informacje na jakim etapie jest jego zwrot.
  - Obiekty- użytkownik, zwrot, status zwrotu

**→ Reklamacja:**

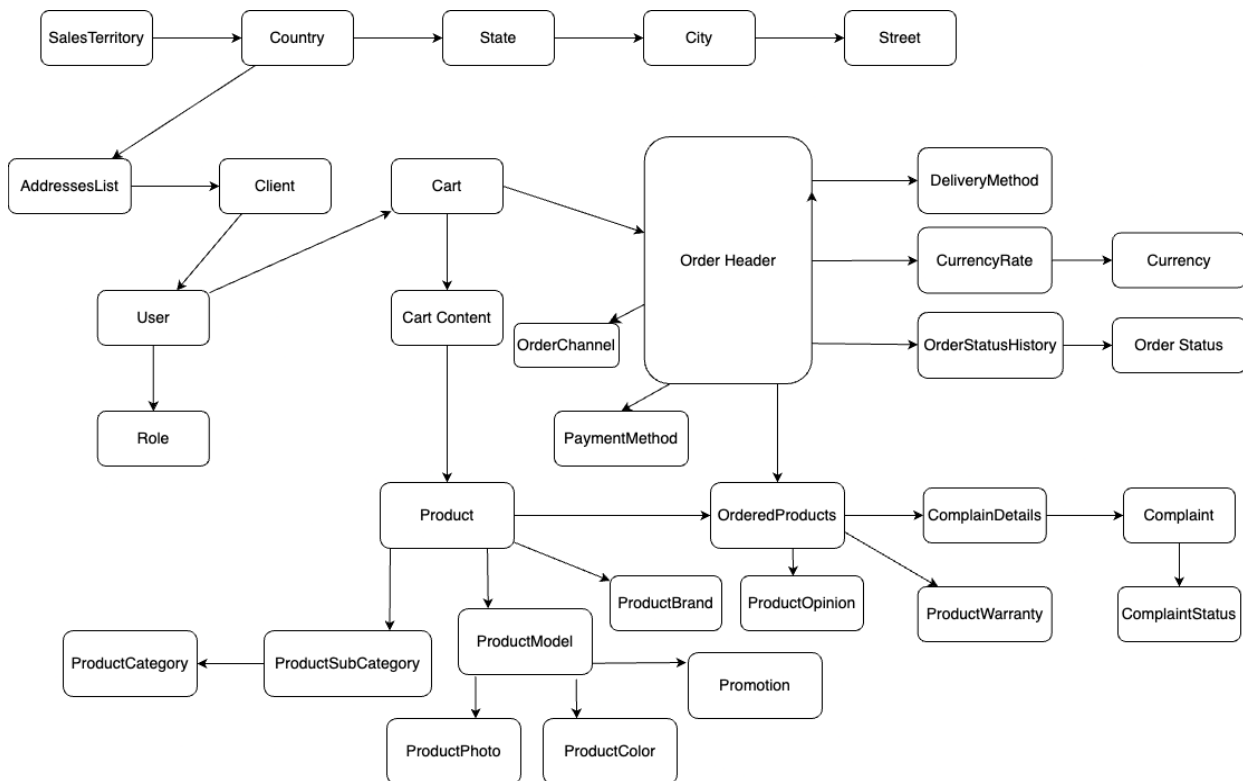
- ◆ Możliwość reklamacji - klient w określonym czasie gwarancyjnym danego produktu, który został przez niego zamówiony, może dokonać reklamacji.
  - Obiekty- użytkownik, zakupione produkty, reklamacja, zamówienie
- ◆ Przebieg reklamacji - klient może odesłać zakupione produkty, które chce zareklamować. Otrzymuje on w wiadomości e-mail etykietę potrzebną do reklamacji przedmiotu. Jeśli reklamacja zostanie rozpatrzona pozytywnie klient otrzyma nowy produkt lub zwrot środków (w przypadku opłaty przy odbiorze klient musi podać swój numer konta bankowego). Jeżeli reklamacja zostanie rozpatrzona negatywnie, klientowi zostaną odesłane reklamowane przez niego produkty.
  - Obiekty- użytkownik, zakupione produkty, reklamacja, metoda płatności

- ◆ Status reklamacji - klient ma możliwość obserwowania aktualnego statusu procesu reklamacji, po wybraniu zakładki “status” w menu reklamacji. Ma on wtedy informacje na jakim etapie jest jego reklamacja.
  - Obiekty- użytkownik, reklamacja, status reklamacji





## ROZDZIAŁ 4. MODEL KONCEPTUALNY PROJEKTOWANEJ BAZY DANYCH



### Opis encji:

1. AddressList- jest to lista preferowanych adresów danego klienta.
2. Cart- są to wszystkie koszyki klientów.
3. CartContent- zawartość danego koszyka.
4. City- określa miasto.
5. Client- osoba fizyczna posiadająca w naszym sklepie.
6. ComplaintDetails- szczegóły zwrotu lub reklamacji.
7. Complaint- reklamacja lub zwrot zamówionych produktów
8. ComplaintStatus- opisuje możliwe statusy procesu reklamacji lub zwrotu dla danego zamówienia.
9. Country- określa kraj.
10. Currency- dostępne waluty, którymi można opłacić dane zamówienie.
11. CurrencyRate- kurs walut, który występuje w danym dniu.
12. DeliveryMethod- dostępne metody dostawy dla zamówień.
13. OrderedProducts- produkty zamówione przez użytkownika w ramach danego zamówienia.
14. OrderHeader- szczegóły zamówień dokonanych przez użytkowników.
15. OrderStatus- możliwe statusy jakie może przyjąć zamówienie.
16. OrderStatusHistory- statusy przyjmowane przez konkretne zamówienie.
17. PaymentMethod- możliwe metody płatności jakimi użytkownik może opłacić dane zamówienie.
18. Product- produkty dostępne w sklepie.

19. ProductBrand- marka danego modelu produktu.
20. ProductCategory- kategoria w której jest dany produkt.
21. ProductColor- kolor danego modelu produktu.
22. ProductModel- modele danych produktów.
23. ProductOpinion- opinie wystawione przez użytkowników, dotyczące zamówionych produktów.
24. ProductPhoto- zdjęcia danych modeli produktów.
25. ProductSubcategory- podkategoria do której należy dany produkt.
26. ProductWarranty- gwarancja na zakupiony produkt.
27. Promotion- promocja, która trwa pewien okres, dotyczy konkretnej ilości danego modelu produktu. Polega na czasowym obniżeniu cen produktów.
28. Role- rola użytkownika w sklepie, definiuje jego uprawnienia.
29. SalesTerritory- obszar na którym obowiązuje sprzedaż danych produktów.
30. State-region, województwo w danym kraju.
31. Street- ulica w danym mieście.
32. User- jest to konto osoby fizycznej.
33. OrderChannel- jest to metoda dzięki której użytkownik może dokonać zakupu w sklepie (aplikacja mobilna, strona internetowa sklepu).

**Związki pomiędzy poszczególnymi encjami:**

1. *SalesTerritory* - *Country*: Związek jeden-do-wielu (1:N)
2. *Country* - *State*: Związek jeden-do-wielu (1:N)
3. *Role*-*User*: Związek jeden-do-wielu (1:N)
4. *Client* - *AddressList*: Związek jeden-do-wielu (1:N)
5. *Cart* - *CartContent*: Związek jeden-do-wielu (1:N)
6. *ProductModel*-*Product*: Związek jeden-do-wielu (1:N)
7. *Complaint* - *ComplaintDetails*: Związek jeden-do-wielu (1:N)
8. *Promotion* - *ProductModel*: Związek wiele-do-wielu (M:N)
9. *User* - *Cart*: Związek jeden-do-wielu (1:N)
10. *Cart* - *OrderHeader*: Związek jeden-do-jeden (1:1)
11. *DeliveryMethod*-*OrderHeader*: Związek jeden-do-wielu (1:N)
12. *PaymentMethod*-*OrderHeader*: Związek jeden-do-wielu (1:N)
13. *OrderHeader* - *OrderStatusHistory*: Związek jeden-do-wielu (1:N)
14. *OrderStatus* - *OrderStatusHistory*: Związek jeden-do-wielu (1:N)
15. *OrderHeader* - *OrderedProducts*: Związek jeden-do-wielu (1:N)
16. *ProductSubcategory*-*Product*: Związek jeden-do-wielu (1:N)
17. *ProductCategory*-*Subcategory*: Związek jeden-do-wielu (1:N)
18. *ProductBrand*-*Product*: Związek jeden-do-wielu (1:N)
19. *ProductColor*-*ProductModel*: Związek jeden-do-wielu (1:N)
20. *OrderedProducts*-*ProductOpinion*: Związek wiele-do-wielu (N:M)
21. *ProductOrdered* - *ProductWarranty*: Związek jeden-do-jednego (1:1)
22. *ComplaintStatus* - *Complaint*: Związek jeden-do-wielu (1:N)
23. *Product* - *OrderedProducts*: Związek jeden-do-wielu (1:N)
24. *City* - *Street*: Związek jeden-do-wielu (1:N)
25. *State* - *City*: Związek jeden-do-wielu (1:N)



- 26. *CurrencyRate - OrderHeader*: Związek jeden-do-wielu (1:N)
- 27. *Currency - CurrencyRate*: Związek jeden-do-wielu (1:N)
- 28. *Client-User*: Związek jeden-do-jednego (1:1)
- 29. *Product - CartContent*: Związek jeden-do-wielu (1:N)
- 30. *ComplaintDetails - OrderedProducts*: Związek jeden-do-jednego (1:1)
- 31. *ProductModel - ProductPhoto*: Związek jeden-do-wielu (1:N)
- 32. *Country - AddressList*: Związek jeden-do-wielu (1:N)
- 33. *OrderChannel-OrderHeader*: Związek jeden-do-wielu (1:N)

## ROZDZIAŁ 5. MODEL ZWIĄZKÓW ENCJI PROJEKTOWANEJ BAZY DANYCH

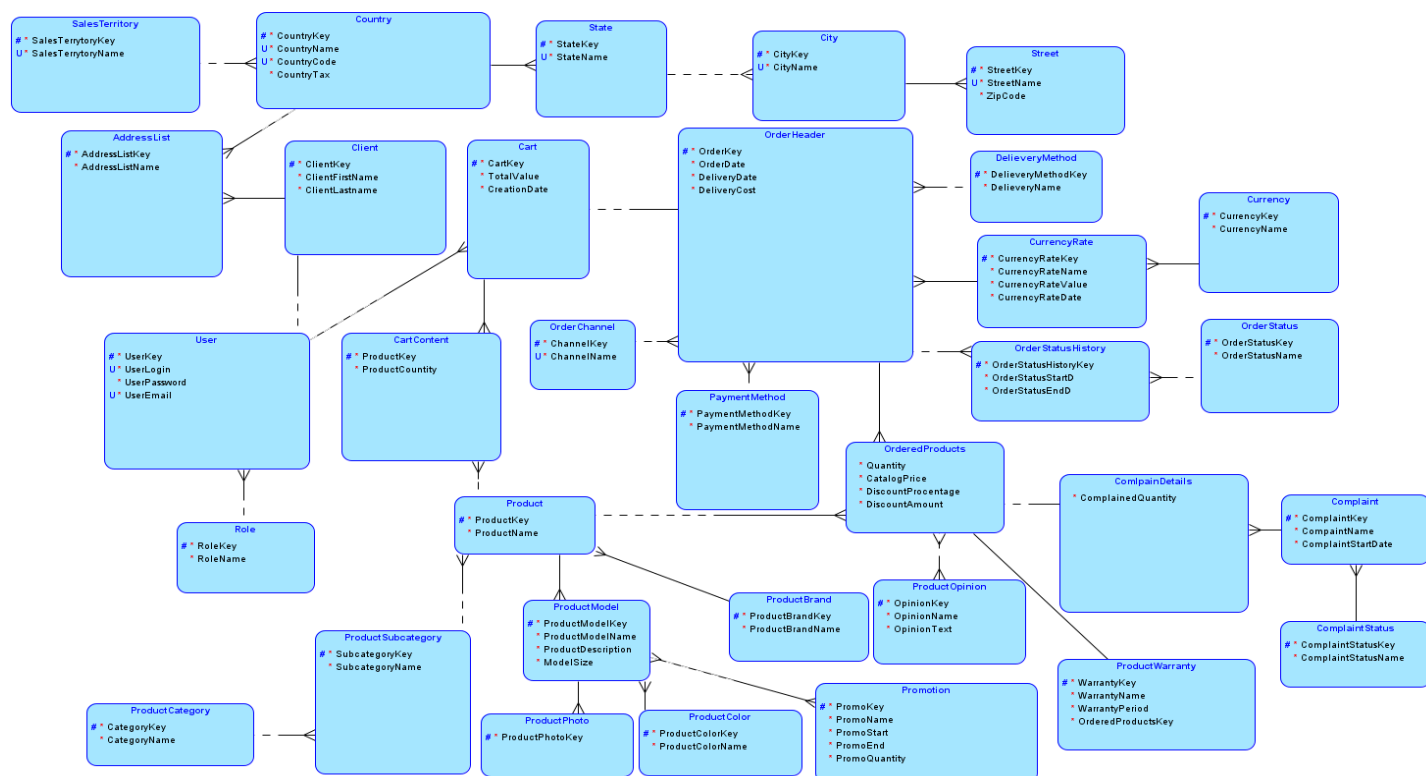


Tabela 1 Specyfikacja atrybutów encji SalesTerritory

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie )	Wartość unikatowa(Tak/Nie )	Opis atrybutu
SalesTerritoryKey	NUMERIC (5)	Tak	Tak	Identyfikator określający
SalesTerritoryName	VARCHAR(40)	Tak	Tak	Nazwa określająca

Tabela 2 Specyfikacja atrybutów encji Country

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
CountryKey	NUMERIC(2)	Tak	Tak	Identyfikator kraju.
CountryName	VARCHAR(50)	Tak	Tak	Nazwa kraju
CountryCode	CHAR(2)	Tak	Tak	Kod służący do identyfikacji danego kraju.
CountryTax	DOUBLE(2)	Tak	Nie	Podatek obowiązujący na terenie danego kraju.

Tabela 3 Specyfikacja atrybutów encji State

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
StateKey	NUMERIC(5)	Tak	Tak	Identyfikator określający dany region.
StateName	VARCHAR(50)	Tak	Tak	Nazwa regionu.

Tabela 4 Specyfikacja atrybutów encji City

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
CityKey	NUMERIC(6)	Tak	Tak	Identyfikator określający dane miasto.
CityName	VARCHAR(50)	Tak	Tak	Nazwa miasta.

Tabela 5 Specyfikacja atrybutów encji Street

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
StreetKey	NUMERIC(10)	Tak	Tak	Identyfikator określający daną ulicę.
StreetName	VARCHAR(50)	Tak	Tak	Nazwa ulicy
ZipCode	CHAR(5)	Tak	Nie	Kod pocztowy obowiązujący na danej ulicy.



Tabela 6 Specyfikacja atrybutów encji AddressList

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
AddressListKey	NUMERIC(10)	Tak	Tak	Identyfikator określający listę adresów podanych przez klienta w sklepie.
AddressListName	VARCHAR(50)	Tak	Nie	Nazwa adresów klienta.

Tabela 7 Specyfikacja atrybutów encji Client

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ClientKey	NUMERIC(10)	Tak	Tak	Identyfikator opisujący klienta sklepu.
ClientFirstName	VARCHAR(50)	Tak	Nie	Imię klienta sklepu
ClientLastName	VARCHAR(50)	Tak	Nie	Nazwisko klienta sklepu

Tabela 8 Specyfikacja atrybutów encji User

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
UserKey	NUMERIC(10)	Tak	Tak	Identyfikator użytkownika posiadającego konto w sklepie.
UserLogin	VARCHAR(16)	Tak	Tak	Login używany przez użytkownika podczas logowania się do systemu.
UserPassword	VARCHAR(30)	Tak	Nie	Hasło używane przez użytkownika podczas logowania się do systemu.
UserEmail	VARCHAR(40)	Tak	Tak	Email używany przez użytkownika podczas logowania się do systemu.

Tabela 9 Specyfikacja atrybutów encji Role

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
RoleKey	NUMERIC(10)	Tak	Tak	Identyfikator opisujący rolę użytkownika w systemie sklepu.
RoleName	VARCHAR(40)	Tak	Nie	Nazwa roli, którą posiada użytkownik sklepu





Tabela 10 Specyfikacja atrybutów encji Cart

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
CartKey	NUMERIC(10)	Tak	Tak	Identyfikator określający pojedynczy koszyk utworzony przez klienta sklepu.
TotalValue	NUMERIC(8, 2)	Tak	Nie	Aktualna wartość wszystkich produktów znajdujących się w danym koszyku.
CreationDate	Date	Tak	Nie	Data utworzenia koszyka.

Tabela 11 Specyfikacja atrybutów encji OrderHeader

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
OrderKey	NUMERIC(10)	Tak	Tak	Identyfikator określający dane zamówienie zrealizowane w sklepie.
OrderDate	Date	Tak	Nie	Data złożenia zamówienia.
DeliveryDate	Date	Tak	Nie	Data doręczenia zamówienia.
DeliveryCost	NUMERIC(10, 2)	Tak	Nie	Koszt wysyłki.

Tabela 12 Specyfikacja atrybutów encji DelieveryMethod

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
DelieveryMethodKey	NUMERIC(2)	Tak	Tak	Identyfikator określający metodę dostawy wybraną dla danego zamówienia.
DelieveryName	VARCHAR(40)	Tak	Nie	Nazwa metody zamówienia.

Tabela 13 Specyfikacja atrybutów encji CurrencyRate

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
CurrencyRateKey	NUMERIC(10)	Tak	Tak	Identyfikator określający kurs obowiązujący dla danej waluty.
CurrencyRateName	VARCHAR(40)	Tak	Nie	Nazwa kursu waluty.
CurrencyRateValue	NUMERIC(8,4)	Tak	Nie	Wartość danego kursu waluty, po której klient dokonuje zakupu.
CurrencyRateDate	Date	Tak	Nie	Data opisująca kurs dla danej waluty w danym dniu.

Tabela 14 Specyfikacja atrybutów encji Currency

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
CurrencyKey	NUMERIC(10)	Tak	Tak	Identyfikator określający walutę użytą w danym zamówieniu.
CurrencyName	VARCHAR(40)	Tak	Nie	Nazwa waluty znajdującej się w sklepie.



Tabela 15 Specyfikacja atrybutów encji OrderStatusHistory

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
OrderStatusHistoryKey	NUMERIC(10)	Tak	Tak	Identyfikator określający status dla danego zamówienia.
OrderStatusStartD	Date	Tak	Nie	Data rozpoczęcia statusu dla konkretnego zamówienia.
OrderStatusEndD	Date	Tak	Nie	Data zakończenia statusu dla konkretnego zamówienia.

Tabela 16 Specyfikacja atrybutów encji OrderStatus

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
OrderStatusKey	NUMERIC(2)	Tak	Tak	Identyfikator określający aktualny status dla danego zamówienia
OrderStatusName	VARCHAR(30)	Tak	Nie	Nazwa statusu danego zamówienia w sklepie.



Tabela 17 Specyfikacja atrybutów encji PaymentMethod

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
PaymentMethodKey	NUMERIC(2)	Tak	Tak	Identyfikator określający jaką metodę płatności użyto do opłacenia danego koszyka.
PaymentMethodName	VARCHAR(30)	Tak	Nie	Nazwa metody płatności znajdującej się w sklepie.

Tabela 18 Specyfikacja atrybutów encji CartContent

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ProductQuantity	NUMERIC(3)	Tak	Nie	Liczba sztuk danego produktu znajdującego się w danym koszyku.

Tabela 19 Specyfikacja atrybutów encji Product

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ProductKey	NUMERIC(10)	Tak	Tak	Identyfikator określający dany produkt znajdujący się w sklepie.
ProductName	VARCHAR(50)	Tak	Nie	Nazwa produktu

Tabela 20 Specyfikacja atrybutów encji ProductSubcategory

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
SubcategoryKey	NUMERIC(2)	Tak	Tak	Identyfikator określający podkategorię dla danej kategorii.
SubcategoryName	VARCHAR(40)	Tak	Nie	Nazwa podkategorii.

Tabela 21 Specyfikacja atrybutów encji ProductCategory

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
CategoryKey	NUMERIC(2)	Tak	Tak	Identyfikator określający kategorię, do której należy dany produkt.
CategoryName	VARCHAR(40)	Tak	Nie	Nazwa kategorii produktowej w sklepie.

Tabela 22 Specyfikacja atrybutów encji ProductModel

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ProductModelKey	NUMERIC(10)	Tak	Tak	Identyfikator określający model danego produktu
ProductModelName	VARCHAR(40)	Tak	Nie	Nazwa modelu danego produktu
ProductDescription	VARCHAR(1000)	Tak	Nie	Opis zawierający informację o produkcie znajdującym się w sklepie
ModelSize	NUMERIC(2)	Tak	Nie	Rozmiarówka danego modelu.

Tabela 23 Specyfikacja atrybutów encji ProductPhoto

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ProductPhotoKey	NUMERIC(10)	Tak	Tak	Identyfikator określający zdjęcie produktu znajdującego się w sklepie

Tabela 24 Specyfikacja atrybutów encji ProductColor

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ProductColorKey	NUMERIC(3)	Tak	Tak	Identyfikator koloru danego produktu znajdującego się w sklepie.
ProductColorName	VARCHAR(40)	Tak	Nie	Nazwa koloru produktu.



Tabela 25 Specyfikacja atrybutów encji Promotion

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
PromoKey	NUMERIC(10)	Tak	Tak	Identyfikator określający pojedynczą promocję w sklepie.
PromoName	VARCHAR(40)	Tak	Nie	Nazwa promocji
PromoStart	Date	Tak	Nie	Data rozpoczęcia promocji
PromoEnd	Date	Tak	Nie	Data zakończenia promocji
PromoQuantity	NUMERIC(4)	Tak	Nie	Liczba sztuk danego produktu objętego promocją

Tabela 26 Specyfikacja atrybutów encji ProductBrand

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ProductBrandKey	NUMERIC(5)	Tak	Tak	Identyfikator określający markę produktu.
ProductBrandName	VARCHAR(40)	Tak	Nie	Nazwa marki produktu



Tabela 27 Specyfikacja atrybutów encji OrderedProducts

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
Quantity	NUMERIC(3)	Tak	Nie	Liczba sztuk produktów zakupionych w sklepie.
CatalogPrice	NUMERIC(7,2)	Tak	Nie	Cena katalogowa produktu
DiscountProcentage	NUMERIC(7,2)	Tak	Nie	Procent zniżki przyznany automatycznie za zakup danego produktu.
DiscountAmount	NUMERIC(7,2)	Tak	Nie	Kwota rabatu przyznana automatycznie za zakup danego produktu.



Tabela 28 Specyfikacja atrybutów encji ProductOpinion

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
OpinionKey	NUMERIC(10)	Tak	Tak	Identyfikator opisujący opinię użytkownika sklepu o danym produkcie, który został przez niego zamówiony.
OpinionName	VARCHAR(40)	Tak	Nie	Nazwa opinii wystawionej przez użytkownika sklepu
OpinionText	VARCHAR(400 )	Tak	Nie	Treść opinii, która jest wystawiona przez użytkownika, który zakupił produkt.

Tabela 29 Specyfikacja atrybutów encji ComplaintDetails

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie )	Wartość unikatowa(Tak/Nie )	Opis atrybutu
ComplaintQuantity	NUMERIC(10)	Tak	Nie	Liczba sztuk danego produktu, który zostaje poddany procesowi reklamacji.

Tabela 30 Specyfikacja atrybutów encji Complaint

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ComplaintKey	NUMERIC(10)	Tak	Tak	Identyfikator opisujący pojedynczą reklamację zakupionego produktu przez klienta.
ComplaintStartDate	Date	Tak	Nie	Data rozpoczęcia procesu reklamacji.
OrderID	NUMERIC(10)	Tak	Nie	Identyfikator opisujący numer zamówienia danego koszyka.
ComplaintStatusKey	NUMERIC(2)	Tak	Nie	Identyfikator określający aktualny status danego procesu reklamacyjnego.



Tabela 31 Specyfikacja atrybutów encji ComplaintStatus

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ComplaintStatusKey	NUMERIC(2)	Tak	Tak	Identyfikator określający status danego procesu reklamacyjnego.
ComplaintStatusName	VARCHAR(40)	Tak	Nie	Nazwa statusu procesu reklamacyjnego.



Tabela 32 Specyfikacja atrybutów encji ProductWarranty

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
WarrantyKey	NUMERIC(10)	Tak	Tak	Identyfikator określający proces gwarancyjny produktu zakupionego w sklepie.
WarrantyName	VARCHAR(40)	Tak	Nie	Nazwa gwarancji
WarrantyPeriod	NUMERIC(5)	Tak	Nie	Okres obowiązywania gwarancji
OrderedProductsID	NUMERIC(10)	Tak	Nie	Identyfikator określający produkty, które zostały zakupione w sklepie.

Tabela 33 Specyfikacja atrybutów encji OrderChannel

Nazwa atrybutu	Typ danych i maksymalny rozmiar	Wartość wymagana(Tak/Nie)	Wartość unikatowa(Tak/Nie)	Opis atrybutu
ChannelKey	NUMERIC(2)	Tak	Tak	Identyfikator określający kanał sprzedaży użyty podczas składania zamówienia..
ChannelName	VARCHAR(40)	Tak	Tak	Nazwa kanału sprzedaży używanego podczas składania zamówienia w sklepie,.

**Szczegółowy opis relacji pomiędzy encjami:**

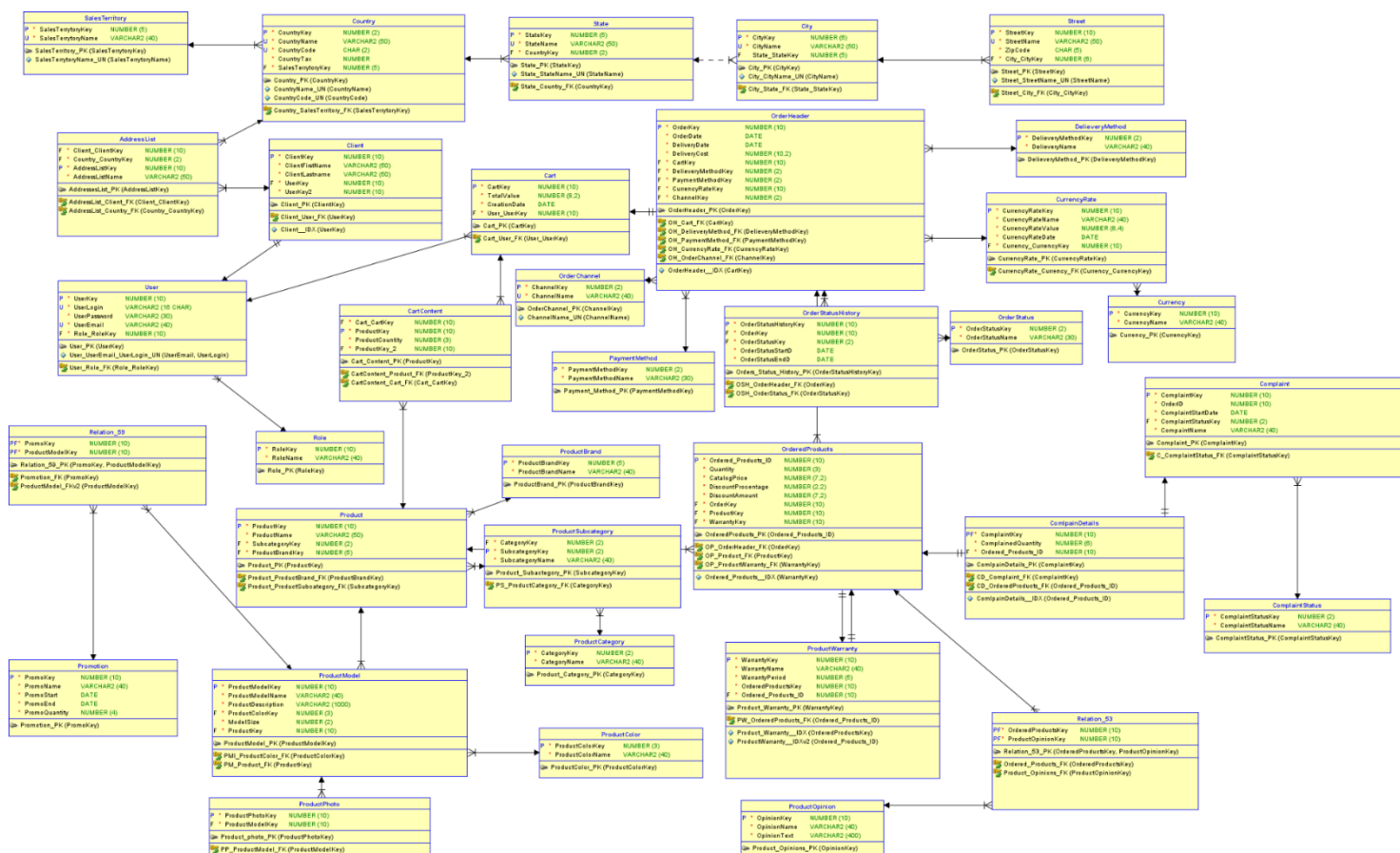
1. *SalesTerritory - Country*: Związek jeden-do-wielu (1:N), gdzie jeden obszar sprzedaży może obejmować wiele krajów, ale każdy kraj jest przypisany do jednego obszaru sprzedaży.
2. *Country - State*: Związek jeden-do-wielu (1:N), oznaczający, że jeden kraj może zawierać wiele stanów, każdy stan musi należeć do dokładnie jednego kraju.
3. *Role-User*: Związek jeden-do-wielu (1:N), oznaczający, że jedna rola może być przypisana do wielu użytkowników, każdy użytkownik musi mieć przypisaną dokładnie jedną rolę.
4. *Client - AddressList*: Związek jeden-do-wielu (1:N), gdzie jeden klient musi mieć przynajmniej jeden adres, każdy adres musi być przypisany do przynajmniej jednego klienta.
5. *Cart - CartContent*: Związek jeden-do-wielu (1:N), gdzie jeden koszyk musi zawierać przynajmniej jedną zawartość koszyka, każda zawartość koszyka musi znajdować się w dokładnie jednym koszyku.
6. *Product-ProductModel*: Związek jeden-do-wielu (1:N), jeden produkt musi posiadać przynajmniej jeden model, jeden model musi należeć do dokładnie jednego produktu.

7. *Complaint - ComplaintDetails*: Związek jeden-do-wielu (1:N), każdy zwrot musi mieć przynajmniej jedno szczegóły zwrotu, poszczególne szczegóły zwrotu muszą być przyporządkowane do dokładnie jednego zwrotu
8. *Promotion - ProductModel*: Związek wiele-do-wielu (M:N), gdzie promocje muszą obejmować przynajmniej jeden model produktu, każdy produkt może należeć do wielu promocji.
9. *User - Cart*: Związek jeden-do-wielu (1:N), wskazujący, że jeden użytkownik może mieć wiele koszyków, ale każdy koszyk jest przypisany do jednego użytkownika.
10. *Cart - OrderHeader*: Związek jeden-do-jeden (1:1), każdy koszyk może zostać zamówiony, każde zamówienie musi być w dokładnie jednym koszyku.
11. *DeliveryMethod-OrderHeader*: Związek jeden-do-wielu (1:N), każda metoda dostawy może być przypisana do wielu nagłówków zamówienia, każdy nagłówek zamówienia musi mieć przypisaną dokładnie jedną metodę dostawy.
12. *PaymentMethod-OrderHeader*: Związek jeden-do-wielu (1:N), jedna metoda płatności może wystąpić w wielu nagłówkach zamówienia, w jednym nagłówku zamówienia musi być dokładnie jedna metoda płatności.
13. *OrderHeader - OrderStatusHistory*: Związek jeden-do-wielu (1:N), jedno zamówienie może mieć wiele statusów w swojej historii, status musi być przypisany do zamówienia.
14. *OrderStatus - OrderStatusHistory*: Związek jeden-do-wielu (1:N), jeden status może być przypisany do wielu historii statusów, w jednej historii statusów musi być zawarty przynajmniej jeden status.
15. *OrderHeader - OrderedProducts*: Związek jeden-do-wielu (1:N), gdzie jedno zamówienie musi zawierać przynajmniej jeden zamówiony produkt, zamówiony produkt musi należeć do dokładnie jednego zamówienia.
16. *ProductSubcategory-Product*: Związek jeden-do-wielu (1:N), gdzie jeden produkt musi należeć do dokładnie jednej podkategorii, ale podkategoria może zawierać wiele produktów.
17. *ProductCategory-Subcategory*: Związek jeden-do-wielu (1:N), gdzie jedna jedna kategoria może należeć do wielu podkategorii, jedna podkategoria musi należeć do dokładnie jednej kategorii.
18. *ProductBrand-Product*: Związek jeden-do-wielu (1:N), gdzie jedna marka musi być przypisana do co najmniej jednego produktu, jeden produkt musi być sygnowany dokładnie jedną marką.
19. *ProductColor-ProductModel*: Związek jeden-do-wielu (1:N), jeden kolor dominujący musi występować w co najmniej jednym modelu produktu, jeden model produktu musi mieć przyporządkowany dokładnie jeden kolor dominujący.
20. *OrderedProducts-ProductOpinion*: Związek wiele-do-wielu (N:M), co oznacza, każdy zamówiony produkt może mieć wiele opinii, każda opinia musi dotyczyć przynajmniej jednego zamówionego modelu danego produktu.
21. *ProductOrdered - ProductWarranty*: Związek jeden-do-jednego (1:1), gdzie każdy zamówiony produkt ma określoną dokładnie jedną gwarancję.
22. *ComplaintStatus - Complaint*: Związek jeden-do-wielu (1:N), jeden status zwrotu musi wystąpić w przynajmniej jednym zwrocie, w jednym zwrocie występuje przynajmniej jeden status.





- 23. *Product - OrderedProducts*: Związek jeden-do-wielu (1:N), jeden produkt może być zamówiony wielokrotnie, w zamówionych produktach musi znajdować się co najmniej jeden produkt .
- 24. *City - Street*: Związek jeden-do-wielu (1:N), gdzie jedno miasto musi zawierać przynajmniej jedną ulicę, jedna ulica musi należeć do dokładnie jednego miasta.
- 25. *State - City*: Związek jeden-do-wielu (1:N), gdzie jeden region może zawierać wiele miast, jedno miasto może należeć do dokładnie jednego regionu.
- 26. *CurrencyRate - OrderHeader*: Związek jeden-do-wielu (1:N), gdzie jeden kurs z danego dnia musi wystąpić w przynajmniej jednym zamówieniu, w szczegółach jednego zamówienia musi wystąpić dokładnie jeden kurs waluty z danego dnia.
- 27. *Currency - CurrencyRate*: Związek jeden-do-wielu (1:N), gdzie jedna waluta musi wystąpić w przynajmniej jednym kursie waluty z danego dnia, pojedynczy kurs danej waluty z danego dnia musi zawierać dokładnie jedną walutę.
- 28. *Client-User*: Związek jeden-do-jednego (1:1), gdzie jeden użytkownik może być klientem sklepu internetowego, każdy klient musi być użytkownikiem serwisu.
- 29. *Product - CartContent*: Związek jeden-do-wielu (1:N), gdzie jeden produkt może być częścią wielu zawartości koszyków, jedna zawartość koszyka musi zawierać co najmniej jeden produkt.
- 30. *ComplaintDetails - OrderedProducts*: Związek jeden-do-jednego (1:1), gdzie jeden zamówiony produkt może zostać zwrócony, każdy zwrot musi dotyczyć dokładnie jednego zamówionego produktu.
- 31. *ProductModel - ProductPhoto*: Związek jeden-do-wielu (1:N), gdzie jeden model produktu musi mieć przynajmniej jedno przypisane zdjęcie, jedno zdjęcie jest przypisane dokładnie jednemu modelowi produktu.
- 32. *Country - AddressList*: Związek jeden-do-wielu (1:N), gdzie jeden adres może być na wielu listach adresowych, jedna lista adresów musi zawierać co najmniej jeden adres.
- 33. *OrderChannel-OrderHeader*: Związek jeden-do-wielu (1:N), gdzie jeden kanał może być użyty podczas składania wielu zamówień, konkretne zamówienie ma dokładnie jeden kanał sprzedaży.



Nazwa tabeli	Nazwa kolumny	Nazwa więzów
SalesTerritory	SalesTerrytoryKey	SalesTerritory_PK
Country	CountryKey	Country_PK
State	StateKey	State_PK
City	CityKey	City_PK
Street	StreetKey	Street_PK
AddressList	AddressListKey	AddressList_PK
Client	ClientKey	Client_PK

User	UserKey	User_PK
Role	RoleKey	Role_PK
Cart	CartKey	Cart_PK
OrderHeader	Orderkey	OrderHeader_PK
DelieveryMethod	DelieveryMethodKey	DelieveryMethod_PK
CurrencyRate	CurrencyRateKey	CurrencyRate_PK
Currency	CurrencyKey	Currency_PK
OrderStatusHistory	OrderStatusHistoryKey	OrderStatusHistory_PK
OrderStatus	OrderStatusKey	OrderStatus_PK
Product	ProductKey	Product_PK
ProductSubcategory	SubcategoryKey	Product_Subcategory_PK
ProductCategory	CategoryKey	Product_Category_PK
ProductModel	ProductModelKey	ProductModel_PK
ProductPhoto	ProductPhotoKey	Product_photo_PK
ProductColor	ProductColorKey	ProductColor_PK
Promotion	PromoKey	Promotion_PK
ProductOpinion	OpinionKey	Product_Opinions_PK
ProductBrand	ProductBrandKey	ProductBrand_PK
Complaint	ComplaintKey	Complaint_PK
ComplaintStatus	ComplaintStatusKey	ComplaintStatus_PK
ProductWarranty	WarrantyKey	Product_Warranty_PK
PaymentMethod	PaymentMethodKey	Payment_Method_PK
OrderChannel	OrderChannelKey	OrderChannel_PK



Tabela 2 Definicja relacji między poszczególnymi tabelami bazy danych

Tabela pierwotna	Tabela referencyjna	Nazwa kolumny (Foreign key)	Nazwa więzów
OrderHeader	DelieveryMethod	DelieveryMethodKey	OH_DelieveryMethod_FK
OrderHeader	CurrencyRate	CurrencyRateKey	OH_PaymentMethod_FK
OrderHeader	PaymentMethod	PaymentMethodKey	OH_PaymentMethod_FK
OrderHeader	Cart	CartKey	OH_Cart_FK
OrderHeader	OrderChannel	OrderChannelKey	OH_OrderChannel_FK
CurrencyRate	Currency	CurrencyKey	CurrencyRate_Currency_FK
OrderStatusHistory	OrderStatus	OrderStatusKey	OSH_OrderStatus_FK
OrderStatusHistory	OrderHeader	OrderKey	OSH_OrderHeader_FK
ComplainDetails	Complaint	ComplaintKey	CD_Complaint_FK
ComplainDetails	OrderedProducts	Ordered_Products_ID	CD_OrderedProducts_FK
Complaint	ComplaintStatus	ComplaintStatusKey	C_ComplaintStatus_FK
ProductSubcategory	ProductCategory	CategoryKey	PS_ProductCategory_FK
ProductModel	ProductColor	ProductColorKey	PM_ProductColor_FK
Street	City	CityKey	Street_City_FK
City	State	StateKey	City_State_FK



State	Country	CountryKey	State_Country_FK
Country	SalesTerritory	SalesTerritoryKey	Country_SalesTerritory_FK
User	Role	RoleKey	User_Role_FK
Client	User	UserKey	Client_User_FK
Cart	User	UserKey	Cart_User_FK
ProductModel	Product	ProductModelKey	PM_Product_FK
Product	ProductBrand	ProductBrandKey	Product_ProductBrand_FK
Product	ProductSubcategory	ProductSubcategoryKey	Product_ProductSubcategory_FK
CartContent	Cart	CartKey	CartContent_Cart_FK
CartContent	Product	ProductKey	CartContent_Product_FK
AddressList	Country	CountryKey	AddressList_Country_FK
AddressList	Client	ClientKey	AddressList_Client_FK
ProductPhoto	ProductModel	ProductModelKey	PP_ProductModel_FK
OrderedProducts	ProductWarranty	WarrantyKey	OP_ProductWarranty_FK
OrderedProducts	Product	ProductKey	OP_Product_FK
OrderedProducts	OrderHeader	OrderKey	OP_OrderHeader_FK
ProductWarranty	OrderedProducts	Ordered_Products_ID	PW_OrderedProducts_FK



## **ROZDZIAŁ 7. KOD SQL – TWORZENIE BAZY DANYCH**

```
-- Generated by Oracle SQL Developer Data Modeler 23.1.0.087.0806
-- at:      2023-12-28 18:21:20 CET
-- site:    Oracle Database 11g
-- type:    Oracle Database 11g
```

```
-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
```

```
-- predefined type, no DDL - XMLTYPE
```

```
CREATE TABLE addresslist (
  client_clientkey NUMBER(10) NOT NULL,
  country_countrykey NUMBER(2) NOT NULL,
  addresslistkey NUMBER(10) NOT NULL,
  addresslistname VARCHAR2(50) NOT NULL
);
```

```
ALTER TABLE addresslist ADD CONSTRAINT addresseslist_pk PRIMARY KEY ( addresslistkey );
```

```
CREATE TABLE cart (
  cartkey NUMBER(10) NOT NULL,
  totalvalue NUMBER(8, 2) NOT NULL,
  creationdate DATE NOT NULL,
  user_userkey NUMBER(10) NOT NULL
);
```

```
ALTER TABLE cart ADD CONSTRAINT cart_pk PRIMARY KEY ( cartkey );
```

```
CREATE TABLE cartcontent (
  cart_cartkey NUMBER(10) NOT NULL,
  productkey NUMBER(10) NOT NULL,
  productcountity NUMBER(3) NOT NULL,
  productkey_2 NUMBER(10) NOT NULL
);
```

```
ALTER TABLE cartcontent ADD CONSTRAINT cart_content_pk PRIMARY KEY ( productkey );
```

```
CREATE TABLE city (
  citykey NUMBER(6) NOT NULL,
  cityname VARCHAR2(50) NOT NULL,
  state_statekey NUMBER(5)
);
```

```
ALTER TABLE city ADD CONSTRAINT city_pk PRIMARY KEY ( citykey );
```

```
ALTER TABLE city ADD CONSTRAINT city_cityname_un UNIQUE ( cityname );
```

```
CREATE TABLE client (
  clientkey NUMBER(10) NOT NULL,
  clientfirstname VARCHAR2(50) NOT NULL,
  clientlastname VARCHAR2(50) NOT NULL,
  userkey NUMBER(10) NOT NULL,
  userkey2 NUMBER(10) NOT NULL
);
```

```
CREATE UNIQUE INDEX client__idx ON
  client (
    userkey
  ASC );
```

```
ALTER TABLE client ADD CONSTRAINT client_pk PRIMARY KEY ( clientkey );
```

```
CREATE TABLE complaindetails (
  complaintkey    NUMBER(10) NOT NULL,
  complainedquantity NUMBER(6) NOT NULL,
  ordered_products_id NUMBER(10) NOT NULL
);
```

```
CREATE UNIQUE INDEX complaindetails__idx ON
  complaindetails (
    ordered_products_id
  ASC );
```

```
ALTER TABLE complaindetails ADD CONSTRAINT complaindetails_pk PRIMARY KEY ( complaintkey
);
```

```
CREATE TABLE complaint (
  complaintkey    NUMBER(10) NOT NULL,
  orderid         NUMBER(10) NOT NULL,
  complaintstartdate DATE NOT NULL,
  complaintstatuskey NUMBER(2) NOT NULL,
  complaintname    VARCHAR2(40) NOT NULL
);
```

```
ALTER TABLE complaint ADD CONSTRAINT complaint_pk PRIMARY KEY ( complaintkey );
```

```
CREATE TABLE complaintstatus (
  complaintstatuskey NUMBER(2) NOT NULL,
  complaintstatusname VARCHAR2(40) NOT NULL
);
```

```
ALTER TABLE complaintstatus ADD CONSTRAINT complaintstatus_pk PRIMARY KEY (
complaintstatuskey );
```

```
CREATE TABLE country (
  countrykey    NUMBER(2) NOT NULL,
  countryname    VARCHAR2(50) NOT NULL,
  countrycode    CHAR(2) NOT NULL,
  countrytax     NUMBER NOT NULL,
  salesterritorykey NUMBER(5) NOT NULL
);
```

```
ALTER TABLE country ADD CONSTRAINT country_pk PRIMARY KEY ( countrykey );
```

```
ALTER TABLE country ADD CONSTRAINT countrycode_un UNIQUE ( countrycode );
```

```
ALTER TABLE country ADD CONSTRAINT countryname_un UNIQUE ( countryname );
```

```
CREATE TABLE currency (
  currencykey NUMBER(10) NOT NULL,
  currencyname VARCHAR2(40) NOT NULL
);
```



```
ALTER TABLE currency ADD CONSTRAINT currency_pk PRIMARY KEY ( currencykey );
```

```
CREATE TABLE currencyrate (  
    currencyratekey NUMBER(10) NOT NULL,  
    currencyratename VARCHAR2(40) NOT NULL,  
    currencyratevalue NUMBER(8, 4) NOT NULL,  
    currencyratedate DATE NOT NULL,  
    currency_currencykey NUMBER(10) NOT NULL  
);
```

```
ALTER TABLE currencyrate ADD CONSTRAINT currencyrate_pk PRIMARY KEY ( currencyratekey );
```

```
CREATE TABLE delieverymethod (  
    delieverymethodkey NUMBER(2) NOT NULL,  
    delieveryname VARCHAR2(40) NOT NULL  
);
```

```
ALTER TABLE delieverymethod ADD CONSTRAINT delieverymethod_pk PRIMARY KEY (  
delieverymethodkey );
```

```
CREATE TABLE orderchannel (  
    channelkey NUMBER(2) NOT NULL,  
    channelname VARCHAR2(40) NOT NULL  
);
```

```
ALTER TABLE orderchannel ADD CONSTRAINT orderchannel_pk PRIMARY KEY ( channelkey );
```

```
ALTER TABLE orderchannel ADD CONSTRAINT channelname_un UNIQUE ( channelname );
```

```
CREATE TABLE orderedproducts (  
    ordered_products_id NUMBER(10) NOT NULL,  
    quantity NUMBER(3) NOT NULL,  
    catalogprice NUMBER(7, 2) NOT NULL,  
    discountpercentage NUMBER(2, 2) NOT NULL,  
    discountamount NUMBER(7, 2) NOT NULL,  
    orderkey NUMBER(10) NOT NULL,  
    productkey NUMBER(10) NOT NULL,  
    warrantykey NUMBER(10) NOT NULL  
);
```

```
CREATE UNIQUE INDEX ordered_products__idx ON  
orderedproducts (  
    warrantykey  
ASC );
```

```
ALTER TABLE orderedproducts ADD CONSTRAINT orderedproducts_pk PRIMARY KEY (  
ordered_products_id );
```

```
CREATE TABLE orderheader (  
    orderkey NUMBER(10) NOT NULL,  
    orderdate DATE NOT NULL,  
    deliverydate DATE NOT NULL,  
    deliverycost NUMBER(10, 2) NOT NULL,  
    cartkey NUMBER(10) NOT NULL,  
    delieverymethodkey NUMBER(2) NOT NULL,  
    paymentmethodkey NUMBER(2) NOT NULL,  
    currencyratekey NUMBER(10) NOT NULL,
```





```
channelkey    NUMBER(2) NOT NULL
);

CREATE UNIQUE INDEX orderheader__idx ON
  orderheader (
    cartkey
  ASC );

ALTER TABLE orderheader ADD CONSTRAINT orderheader_pk PRIMARY KEY ( orderkey );

CREATE TABLE orderstatus (
  orderstatuskey NUMBER(2) NOT NULL,
  orderstatusname VARCHAR2(30) NOT NULL
);

ALTER TABLE orderstatus ADD CONSTRAINT orderstatus_pk PRIMARY KEY ( orderstatuskey );

CREATE TABLE orderstatushistory (
  orderstatushistorykey NUMBER(10) NOT NULL,
  orderkey              NUMBER(10) NOT NULL,
  orderstatuskey        NUMBER(2) NOT NULL,
  orderstatusstartd     DATE NOT NULL,
  orderstatusendd       DATE NOT NULL
);

ALTER TABLE orderstatushistory ADD CONSTRAINT orders_status_history_pk PRIMARY KEY (
orderstatushistorykey );

CREATE TABLE paymentmethod (
  paymentmethodkey NUMBER(2) NOT NULL,
  paymentmethodname VARCHAR2(30) NOT NULL
);

ALTER TABLE paymentmethod ADD CONSTRAINT payment_method_pk PRIMARY KEY (
paymentmethodkey );

CREATE TABLE product (
  productkey    NUMBER(10) NOT NULL,
  productname   VARCHAR2(50) NOT NULL,
  subcategorykey NUMBER(2) NOT NULL,
  productbrandkey NUMBER(5) NOT NULL
);

ALTER TABLE product ADD CONSTRAINT product_pk PRIMARY KEY ( productkey );

CREATE TABLE productbrand (
  productbrandkey NUMBER(5) NOT NULL,
  productbrandname VARCHAR2(40) NOT NULL
);

ALTER TABLE productbrand ADD CONSTRAINT productbrand_pk PRIMARY KEY ( productbrandkey );

CREATE TABLE productcategory (
  categorykey NUMBER(2) NOT NULL,
  categoryname VARCHAR2(40) NOT NULL
);
```



```
ALTER TABLE productcategory ADD CONSTRAINT product_category_pk PRIMARY KEY ( categorykey );
```

```
CREATE TABLE productcolor (  
    productcolorkey NUMBER(3) NOT NULL,  
    productcolorname VARCHAR2(40) NOT NULL  
);
```

```
ALTER TABLE productcolor ADD CONSTRAINT productcolor_pk PRIMARY KEY ( productcolorkey );
```

```
CREATE TABLE productmodel (  
    productmodelkey NUMBER(10) NOT NULL,  
    productmodelname VARCHAR2(40) NOT NULL,  
    productdescription VARCHAR2(1000) NOT NULL,  
    productcolorkey NUMBER(3) NOT NULL,  
    modelsize NUMBER(2) NOT NULL,  
    productkey NUMBER(10) NOT NULL  
);
```

```
ALTER TABLE productmodel ADD CONSTRAINT productmodel_pk PRIMARY KEY ( productmodelkey );
```

```
CREATE TABLE productopinion (  
    opinionkey NUMBER(10) NOT NULL,  
    opinionname VARCHAR2(40) NOT NULL,  
    opiniontext VARCHAR2(400) NOT NULL  
);
```

```
ALTER TABLE productopinion ADD CONSTRAINT product_opinions_pk PRIMARY KEY ( opinionkey );
```

```
CREATE TABLE productphoto (  
    productphotokey NUMBER(10) NOT NULL,  
    productmodelkey NUMBER(10) NOT NULL  
);
```

```
ALTER TABLE productphoto ADD CONSTRAINT product_photo_pk PRIMARY KEY ( productphotokey );
```

```
CREATE TABLE productsubcategory (  
    categorykey NUMBER(2) NOT NULL,  
    subcategorykey NUMBER(2) NOT NULL,  
    subcategoryname VARCHAR2(40) NOT NULL  
);
```

```
ALTER TABLE productsubcategory ADD CONSTRAINT product_subactegory_pk PRIMARY KEY ( subcategorykey );
```

```
CREATE TABLE productwarranty (  
    warrantykey NUMBER(10) NOT NULL,  
    warrantyname VARCHAR2(40) NOT NULL,  
    warrantyperiod NUMBER(5) NOT NULL,  
    orderedproductskey NUMBER(10) NOT NULL,  
    ordered_products_id NUMBER(10) NOT NULL  
);
```

```
CREATE UNIQUE INDEX product_warranty__idx ON  
    productwarranty (  
        orderedproductskey
```



ASC );

```
CREATE UNIQUE INDEX productwarranty__idxv2 ON
  productwarranty (
    ordered_products_id
  ASC );
```

```
ALTER TABLE productwarranty ADD CONSTRAINT product_warranty_pk PRIMARY KEY ( warrantykey
);
```

```
CREATE TABLE promotion (
  promokey   NUMBER(10) NOT NULL,
  promoname  VARCHAR2(40) NOT NULL,
  promostart DATE NOT NULL,
  promoend   DATE NOT NULL,
  promoquantity NUMBER(4) NOT NULL
);
```

```
ALTER TABLE promotion ADD CONSTRAINT promotion_pk PRIMARY KEY ( promokey );
```

```
CREATE TABLE relation_53 (
  orderedproductskey NUMBER(10) NOT NULL,
  productopinionkey  NUMBER(10) NOT NULL
);
```

```
ALTER TABLE relation_53 ADD CONSTRAINT relation_53_pk PRIMARY KEY ( orderedproductskey,
                                                                    productopinionkey );
```

```
CREATE TABLE relation_59 (
  promokey   NUMBER(10) NOT NULL,
  productmodelkey NUMBER(10) NOT NULL
);
```

```
ALTER TABLE relation_59 ADD CONSTRAINT relation_59_pk PRIMARY KEY ( promokey,
                                                                    productmodelkey );
```

```
CREATE TABLE role (
  rolekey NUMBER(10) NOT NULL,
  rolename VARCHAR2(40) NOT NULL
);
```

```
ALTER TABLE role ADD CONSTRAINT role_pk PRIMARY KEY ( rolekey );
```

```
CREATE TABLE salesterritory (
  salesterritorykey NUMBER(5) NOT NULL,
  salesterritoryname VARCHAR2(40) NOT NULL
);
```

```
ALTER TABLE salesterritory ADD CONSTRAINT salesterritory_pk PRIMARY KEY ( salesterritorykey );
```

```
ALTER TABLE salesterritory ADD CONSTRAINT salesterritoryname_un UNIQUE ( salesterritoryname );
```

```
CREATE TABLE state (
  statekey   NUMBER(5) NOT NULL,
  statename  VARCHAR2(50) NOT NULL,
  countrykey NUMBER(2) NOT NULL
);
```



```
ALTER TABLE state ADD CONSTRAINT state_pk PRIMARY KEY ( statekey );

ALTER TABLE state ADD CONSTRAINT state_statename_un UNIQUE ( statename );

CREATE TABLE street (
    streetkey  NUMBER(10) NOT NULL,
    streetname VARCHAR2(50) NOT NULL,
    zipcode   CHAR(5) NOT NULL,
    city_citykey NUMBER(6) NOT NULL
);

ALTER TABLE street ADD CONSTRAINT street_pk PRIMARY KEY ( streetkey );

ALTER TABLE street ADD CONSTRAINT street_streetname_un UNIQUE ( streetname );

CREATE TABLE "User" (
    userkey  NUMBER(10) NOT NULL,
    userlogin VARCHAR2(16 CHAR) NOT NULL,
    userpassword VARCHAR2(30) NOT NULL,
    useremail VARCHAR2(40) NOT NULL,
    role_rolekey NUMBER(10) NOT NULL
);

ALTER TABLE "User" ADD CONSTRAINT user_pk PRIMARY KEY ( userkey );

ALTER TABLE "User" ADD CONSTRAINT user_useremail_userlogin_un UNIQUE ( useremail,
                                                                    userlogin );

ALTER TABLE addresslist
    ADD CONSTRAINT addresslist_client_fk FOREIGN KEY ( client_clientkey )
        REFERENCES client ( clientkey );

ALTER TABLE addresslist
    ADD CONSTRAINT addresslist_country_fk FOREIGN KEY ( country_countrykey )
        REFERENCES country ( countrykey );

ALTER TABLE complaint
    ADD CONSTRAINT c_complaintstatus_fk FOREIGN KEY ( complaintstatuskey )
        REFERENCES complaintstatus ( complaintstatuskey );

ALTER TABLE cart
    ADD CONSTRAINT cart_user_fk FOREIGN KEY ( user_userkey )
        REFERENCES "User" ( userkey );

ALTER TABLE cartcontent
    ADD CONSTRAINT cartcontent_cart_fk FOREIGN KEY ( cart_cartkey )
        REFERENCES cart ( cartkey );

ALTER TABLE cartcontent
    ADD CONSTRAINT cartcontent_product_fk FOREIGN KEY ( productkey_2 )
        REFERENCES product ( productkey );

ALTER TABLE complaintdetails
    ADD CONSTRAINT cd_complaint_fk FOREIGN KEY ( complaintkey )
        REFERENCES complaint ( complaintkey );

ALTER TABLE complaintdetails
    ADD CONSTRAINT cd_orderedproducts_fk FOREIGN KEY ( ordered_products_id )
```

REFERENCES orderedproducts ( ordered\_products\_id );

ALTER TABLE city

ADD CONSTRAINT city\_state\_fk FOREIGN KEY ( state\_statekey )  
REFERENCES state ( statekey );

ALTER TABLE client

ADD CONSTRAINT client\_user\_fk FOREIGN KEY ( userkey )  
REFERENCES "User" ( userkey );

ALTER TABLE country

ADD CONSTRAINT country\_salesterritory\_fk FOREIGN KEY ( salesterritorykey )  
REFERENCES salesterritory ( salesterritorykey );

ALTER TABLE currencyrate

ADD CONSTRAINT currencyrate\_currency\_fk FOREIGN KEY ( currency\_currencykey )  
REFERENCES currency ( currencykey );

ALTER TABLE orderheader

ADD CONSTRAINT oh\_cart\_fk FOREIGN KEY ( cartkey )  
REFERENCES cart ( cartkey );

ALTER TABLE orderheader

ADD CONSTRAINT oh\_currencyrate\_fk FOREIGN KEY ( currencyratekey )  
REFERENCES currencyrate ( currencyratekey );

ALTER TABLE orderheader

ADD CONSTRAINT oh\_delieverymethod\_fk FOREIGN KEY ( delieverymethodkey )  
REFERENCES delieverymethod ( delieverymethodkey );

ALTER TABLE orderheader

ADD CONSTRAINT oh\_orderchannel\_fk FOREIGN KEY ( channelkey )  
REFERENCES orderchannel ( channelkey );

ALTER TABLE orderheader

ADD CONSTRAINT oh\_paymentmethod\_fk FOREIGN KEY ( paymentmethodkey )  
REFERENCES paymentmethod ( paymentmethodkey );

ALTER TABLE orderedproducts

ADD CONSTRAINT op\_orderheader\_fk FOREIGN KEY ( orderkey )  
REFERENCES orderheader ( orderkey );

ALTER TABLE orderedproducts

ADD CONSTRAINT op\_product\_fk FOREIGN KEY ( productkey )  
REFERENCES product ( productkey );

ALTER TABLE orderedproducts

ADD CONSTRAINT op\_productwarranty\_fk FOREIGN KEY ( warrantykey )  
REFERENCES productwarranty ( warrantykey );

ALTER TABLE relation\_53

ADD CONSTRAINT ordered\_products\_fk FOREIGN KEY ( orderedproductskey )  
REFERENCES orderedproducts ( ordered\_products\_id );

ALTER TABLE orderstatushistory

ADD CONSTRAINT osh\_orderheader\_fk FOREIGN KEY ( orderkey )  
REFERENCES orderheader ( orderkey );



```
ALTER TABLE orderstatushistory
  ADD CONSTRAINT osh_orderstatus_fk FOREIGN KEY ( orderstatuskey )
  REFERENCES orderstatus ( orderstatuskey );

ALTER TABLE productmodel
  ADD CONSTRAINT pm_product_fk FOREIGN KEY ( productkey )
  REFERENCES product ( productkey );

ALTER TABLE productmodel
  ADD CONSTRAINT pm_productcolor_fk FOREIGN KEY ( productcolorkey )
  REFERENCES productcolor ( productcolorkey );

ALTER TABLE productphoto
  ADD CONSTRAINT pp_productmodel_fk FOREIGN KEY ( productmodelkey )
  REFERENCES productmodel ( productmodelkey );

ALTER TABLE relation_53
  ADD CONSTRAINT product_opinions_fk FOREIGN KEY ( productopinionkey )
  REFERENCES productopinion ( opinionkey );

ALTER TABLE product
  ADD CONSTRAINT product_productbrand_fk FOREIGN KEY ( productbrandkey )
  REFERENCES productbrand ( productbrandkey );

ALTER TABLE product
  ADD CONSTRAINT product_productsubcategory_fk FOREIGN KEY ( subcategorykey )
  REFERENCES productsubcategory ( subcategorykey );

ALTER TABLE relation_59
  ADD CONSTRAINT productmodel1_fkv2 FOREIGN KEY ( productmodelkey )
  REFERENCES productmodel ( productmodelkey );

ALTER TABLE relation_59
  ADD CONSTRAINT promotion_fk FOREIGN KEY ( promokey )
  REFERENCES promotion ( promokey );

ALTER TABLE productsubcategory
  ADD CONSTRAINT ps_productcategory_fk FOREIGN KEY ( categorykey )
  REFERENCES productcategory ( categorykey );

ALTER TABLE productwarranty
  ADD CONSTRAINT pw_orderedproducts_fk FOREIGN KEY ( ordered_products_id )
  REFERENCES orderedproducts ( ordered_products_id );

ALTER TABLE state
  ADD CONSTRAINT state_country_fk FOREIGN KEY ( countrykey )
  REFERENCES country ( countrykey );

ALTER TABLE street
  ADD CONSTRAINT street_city_fk FOREIGN KEY ( city_citykey )
  REFERENCES city ( citykey );

ALTER TABLE "User"
  ADD CONSTRAINT user_role_fk FOREIGN KEY ( role_rolekey )
  REFERENCES role ( rolekey );

CREATE SEQUENCE orderedproducts_ordered_produc START WITH 1 NOCACHE ORDER;
```

```
CREATE OR REPLACE TRIGGER orderedproducts_ordered_produc BEFORE
  INSERT ON orderedproducts
  FOR EACH ROW
  WHEN ( new.ordered_products_id IS NULL )
BEGIN
  :new.ordered_products_id := orderedproducts_ordered_produc.nextval;
END;
/
```

-- Oracle SQL Developer Data Modeler Summary Report:

```
--
-- CREATE TABLE                35
-- CREATE INDEX                  6
-- ALTER TABLE                 79
-- CREATE VIEW                   0
-- ALTER VIEW                   0
-- CREATE PACKAGE                0
-- CREATE PACKAGE BODY           0
-- CREATE PROCEDURE              0
-- CREATE FUNCTION               0
-- CREATE TRIGGER                1
-- ALTER TRIGGER                 0
-- CREATE COLLECTION TYPE        0
-- CREATE STRUCTURED TYPE        0
-- CREATE STRUCTURED TYPE BODY   0
-- CREATE CLUSTER                0
-- CREATE CONTEXT                0
-- CREATE DATABASE               0
-- CREATE DIMENSION              0
-- CREATE DIRECTORY              0
-- CREATE DISK GROUP              0
-- CREATE ROLE                   0
-- CREATE ROLLBACK SEGMENT        0
-- CREATE SEQUENCE               1
-- CREATE MATERIALIZED VIEW       0
-- CREATE MATERIALIZED VIEW LOG   0
-- CREATE SYNONYM                0
-- CREATE TABLESPACE            0
-- CREATE USER                   0
--
-- DROP TABLESPACE              0
-- DROP DATABASE                 0
--
-- REDACTION POLICY              0
--
-- ORDS DROP SCHEMA              0
-- ORDS ENABLE SCHEMA            0
-- ORDS ENABLE OBJECT            0
--
-- ERRORS                       0
-- WARNINGS                     0
```





**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny







Raport powstał podczas zajęć laboratoryjnych z przedmiotu  
prowadzonego w ramach projektu  
*„Zintegrowany Program Rozwoju Politechniki Lubelskiej – część  
druga”*,  
umowa nr **POWR.03.05.00-00-Z060/18-00**  
w ramach Programu Operacyjnego Wiedza Edukacja Rozwój 2014-  
2020  
współfinansowanego ze środków Europejskiego Funduszu  
Społecznego



**Fundusze Europejskie**  
Wiedza Edukacja Rozwój



**Rzeczpospolita  
Polska**

**Unia Europejska**  
Europejski Fundusz Społeczny

