



**POLITECHNIKA LUBELSKA**

**WYDZIAŁ ELEKTROTECHNIKI I INFORMATYKI**

KIERUNEK STUDIÓW

INFORMATYKA

Przedmiot: Wprowadzenie do systemów baz danych

*Raport z wykonania projektu pt.*

**Tytuł projektu**

Autor:

*Imię i nazwisko studenta*

Lublin, 2020

# Rozdział 1. Opis wybranego obszaru rzeczywistości oraz wskazanie problemu, który zostanie rozwiązany przy pomocy systemu informatycznego wykorzystującego projektowaną bazę danych

Modelowaną rzeczywistość tworzy sklep internetowy, w którym oferowane są różnego rodzaju produkty związane z marką Apple. Sklep prowadzi globalną działalność biznesową, co oznacza, że jego klienci pochodzą z miast zlokalizowanych w krajach znajdujących się na różnych kontynentach. Ze względu na dużą różnorodność klientów, a co za tym idzie możliwą mobilność dokonują oni zakupów nie tylko w macierzystym kraju, ale również podczas pobytu za granicą. Właściciel sklepu dokonał segmentacji krajów według różnych regionów sprzedaży. Oznacza to, że każdy kraj został przypisany do jednego regionu sprzedaży.

W celu ułatwienia klientom dotarcia do określonych produktów podczas wizyty w sklepie internetowym, zostały one sklasyfikowane według różnych kategorii oraz podkategorii produktowych. Każda podkategoria jest przypisana tylko do jednej kategorii, a każdy produkt należy tylko do jednej podkategorii.

Sklep oferuje obecnie swoim klientom różne kanały sprzedaży, takie jak strona webowa (tradycyjna i mobilna) oraz aplikacja mobilna. Składając zamówienie, klient może wybrać jeden z kilku sposobów dostawy bądź odbioru produktu oraz opłacenia zamówienia. Mając swój własny profil w sklepie, klient może monitorować bieżący status jego zamówienia.

Każde zamówienie jest identyfikowane przez unikalny numer. Produkt może być sprzedany po cenie katalogowej bądź też po cenie promocyjnej, wynikającej z faktu otrzymania przez klienta rabatu na ten produkt, wyrażonego w procentach lub kwotą. W zależności od wartości zamówienia, przesyłka może być darmowa lub konieczne jest jej opłacenie przez klienta. Data wysłania zamówionych produktów jest określana indywidualnie dla każdego zamówienia.

# Rozdział 2. Sformułowanie celu budowy systemu informatycznego, wykorzystującego projektowaną bazę danych, oraz prezentacja wymagań funkcjonalnych stawianych systemowi

**Sklep internetowy „Re-Apple”**

System informatyczny nosi nazwę „Re-Apple”. Jego głównym celem jest umożliwienie sprzedaży poprzez stronę webową, a także dostarczenie informacji na temat różnego rodzaju produktów marki Apple. System będzie także dawał możliwość zadania pytania na temat poszczególnego produktu.

**Wymagania funkcjonalne dla klienta**

1. Założenie konta w systemie
2. Możliwość zadania pytania na temat produktu
3. Przeglądanie listy produktów
4. Składanie zamówienia
5. Dokonanie zapłaty
6. Wybranie szczegółów dostawy
7. Podgląd informacji o produktach
8. Podgląd zamówień
9. Śledzenie zamówień
10. Dodawanie oraz usuwanie produktów w koszyku
11. Możliwość zwrotu produktów
12. Zmiana szczegółów dostawy
13. Możliwość anulowania zamówienia

**Wymagania funkcjonalne dla administratora systemu**

1. Logowanie do serwisu
2. Zarządzanie produktami
3. Dodawanie użytkowników systemu
4. Blokowanie dostępu do systemu
5. Odblokowanie dostępu do systemu
6. Zmiana hasła użytkownika
7. Edycja zamówienia
8. Edycja informacji o produkcie
9. Edycja kategorii i podkategorii
10. Odpowiedź na pytania klientów i pracowników

**Wymagania funkcjonalne dla pracownika**

1. Generowanie raportu ze sprzedaży
2. Wysyłanie wiadomości do klienta
3. Złożenie zamówienia w imieniu klienta
4. Przyjęcie zwrotu produktu
5. Podgląd dostępnych towarów w sklepach
6. Odpowiedź na pytania klientów
7. Zadawanie pytań administratorom

# Rozdział 3. Szczegółowy opis procesów realizowanych w wybranej rzeczywistości, zależności między nimi oraz obiektów w nich uczestniczących

**Funkcjonalność klienta**

1. **Założenie konta w systemie**

Klient ma możliwość zarejestrowania się w serwisie podając adres email oraz opcjonalnie numer telefonu oraz ustawieniu hasła. Klient może potwierdzić rejestrację konta za pomocą linku wysłanego na e-mail bądź kodem wysłanym na numer telefonu.

Obiekty rzeczywistości: klient, administrator, konto.

1. **Możliwość zadania pytania na temat produktu**

Klient ma możliwość zadania pytania na temat danego produktu dotyczącego tego produktu, jego wysyłki, bądź dostępności w magazynie/sklepie.

Obiekty rzeczywistości: klient, pracownik, administrator, produkt, pytanie.

1. **Przeglądanie listy produktów**

Klient ma możliwość przeglądu produktów podzielonych na różnego rodzaju kategorie i podkategorie. Możliwe zawężanie grupy produktów poprzez wybór kategorii lub podkategorii lub specyfikacji.

Obiekty rzeczywistości: klient, produkt, kategorie, podkategorie.

1. **Składanie zamówienia**

Klient ma możliwość złożenia zamówienia produktów w koszyku na podany adres zamieszkania, bądź umówić się na odbiór w wybranym punkcie. Klient wybiera również formę płatności (kartą, gotówką przy odbiorze, raty, PayPal).

Obiekty rzeczywistości: klient, zamówienie, adres, płatność.

1. **Dokonanie zapłaty**

Klient może dokonać płatności za pomocą wybranego przez siebie rodzaju płatności.

Obiekty rzeczywistości: klient, zamówienie, płatność.

1. **Wybranie szczegółów dostawy**

Klient może wybrać szczegóły swojego zamówienia takie jak rodzaj odbioru, dostawcę spośród podanych,

Obiekty rzeczywistości: klient, zamówienie, dostawa.

1. **Podgląd informacji o produktach**

Klient może przeglądać informacje zamieszczone na stronie webowej opisujące poszczególne produkty (specyfikacja, opis produktu, zawartość opakowania/produktu).

Obiekty rzeczywistości: klient, produkt.

1. **Podgląd zamówień**

Klient ma możliwość podglądu dokonanych przez siebie zamówień na stronie webowej, ma możliwość sprawdzenia poprawności produktów, ilości oraz ceny.

Obiekty rzeczywistości: klient, zamówienie ?

1. **Śledzenie zamówień**

Klient ma możliwość śledzenia postępu jego zamówienia poprzez wgląd w etapy dokonanego zamówienia, jego ostatnią lokalizację, możliwość odbioru.

Obiekty rzeczywistości: klient, zamówienie, produkt ?

1. **Dodawanie oraz usuwanie produktów w koszyku**

Klient ma możliwość dodawania oraz usuwania poszczególnych produktów z koszyka, zmiany ilości produktów, ostatecznie złożenia zamówienia na produkty w koszyku.

Obiekty rzeczywistości: klient, koszyk, produkty, zamówienie.

1. **Możliwość zwrotu produktów**

Klient ma możliwość dokonania zwrotu zamówionych przez siebie produktów po wyborze powodu zwrotu oraz wypełnieniu formularza. Zwrot musi zostać zatwierdzony przez pracownika następnie klient odzyska wydane pieniądze.

Obiekty rzeczywistości: klient, pracownik, zwrot, produkt.

1. **Zmiana szczegółów dostawy**

Klient ma możliwość zmiany adresu dostawy lub rodzaju odbioru jeśli przesyłka jest na odpowiednim do tego etapie.

Obiekty rzeczywistości: klient, adres, zamówienie.

1. **Możliwość anulowania zamówienia**

Klient ma możliwość anulowania poszczególnego zamówienia. Po anulowaniu dokonany zostanie zwrot pieniędzy jeżeli płatność została wykonana online.

Obiekty rzeczywistości: klient, zamówienie, płatność.

**Funkcjonalność administratora systemu**

1. **Logowanie do serwisu**

Administrator loguje się do systemu specjalnym kontem. Logowanie jest dwuetapowe, drugim etapem jest podanie kodu otrzymanego w wiadomości SMS. W ten sposób administrator uzyskuje pełną funkcjonalność konta.

Obiekty rzeczywistości: administrator, konto.

1. **Zarządzanie produktami**

Administrator ma możliwość edycji przynależności poszczególnych produktów do kategorii i podkategorii.

Obiekty rzeczywistości: administrator, produkt, kategorie, podkategorie.

1. **Dodawanie użytkowników systemu**

Administrator ma możliwość manualnego dodawania użytkowników do systemu w razie problemów klientów czy też zatrudnienia nowych pracowników.

Obiekty rzeczywistości: administrator, konto, klient, pracownik.

1. **Blokowanie dostępu do systemu**

Administrator ma możliwość zablokowania dostępu do systemu poszczególnym klientom na określony czas.

Obiekty rzeczywistości: administrator, klient.

1. **Odblokowanie dostępu do systemu**

Administrator może w dowolnym momencie odblokować system dla poszczególnym klientom.

Obiekty rzeczywistości: administrator, klient.

1. **Zmiana hasła użytkownika**

Administrator może dokonać zmiany hasła dla każdego użytkownika po ówczesnym otrzymaniu zgłoszenia problemu z hasłem.

Obiekty rzeczywistości: administrator, klient, pracownik, konto.

1. **Edycja zamówienia**

Administrator może dokonać edycji zamówienia jeżeli zostanie o to poproszony przez klienta. Może zmienić zamówiony produkt, ilość produktów, ???.

Obiekty rzeczywistości: administrator, zamówienie, produkt.

1. **Edycja informacji o produkcie**

Administrator może zmieniać informacje dotyczące produktu (cena, opis, dostępność).

Obiekty rzeczywistości: administrator, produkt.

1. **Edycja kategorii oraz podkategorii**

Administrator może dodawać oraz usuwać kategorie i podkategorie, edytować już istniejące kategorie i podkategorie.

Obiekty rzeczywistości: administrator, kategorie, podkategorie.

1. **Odpowiedź na pytania klientów i pracowników**

Administrator może odpowiadać na pytania zadawane przez klientów dotyczące produktów, może również odpowiadać na pytania zadawane przez pracowników dotyczące systemu.

Obiekty rzeczywistości: administrator, klient, pracownik, pytanie.

**Funkcjonalność pracownika**

1. **Generowanie raportu ze sprzedaży**

Pracownik ma możliwość wygenerowania raportu ze sprzedaży dokonanych w danym miesiącu.

Obiekty rzeczywistości: pracownik, produkt.

1. **Wysyłanie wiadomości do klienta**

Pracownik ma możliwość wysłania wiadomości do klienta o treści dotyczącej wyłącznie sprzedaży i obsługi produktu lub strony.

Obiekty rzeczywistości: pracownik, klient, produkt, wiadomość.

1. **Złożenie zamówienia w imieniu klienta**

Pracownik ma możliwość złożenia zamówienia produktów wybranych przez klienta. Klient nie musi posiadać konta w systemie żeby pracownik mógł złożyć zamówienie.

Obiekty rzeczywistości: pracownik, klient, produkt.

1. **Przyjęcie zwrotu produktu**

Pracownik ma możliwość przyjęcia zamówionego przez klienta produktu po wypełnieniu przez klienta formularza, przedstawienia potwierdzenia zakupu tego produktu, oraz przeglądu technicznego tego produktu. Pracownik ma prawo odmówić wykonania zwrotu, jeżeli powyższe warunki nie są spełnione.

Obiekty rzeczywistości: pracownik, klient, zwrot, produkt.

1. **Podgląd dostępnych towarów w sklepach**

Pracownik ma możliwość przeglądu produktów znajdujących się w wybranym sklepie

Akcje: 1. Wybór sklepu (Pracownik wybiera docelowy sklep przez wyszukiwanie)

2. Przeglądanie listy dostępnych produktów w sklepie (Pracownik wybiera docelową kategorię/podkategorię i przegląda listę produktów)

Obiekty rzeczywistości: pracownik, sklep, produkt.

1. **Odpowiedź na pytania klientów**

Pracownik ma możliwość dodawania odpowiedzi na zadane przez klienta pytanie.

Obiekty rzeczywistości: pracownik, klient, produkt, pytanie, odpowiedź.

1. **Zadawanie pytań administratorom**

Pracownik może zadawać pytania związane z ogólnym systemem (wprowadzanie kategori i podkategorii, edycje produktów) skierowanych do administratorów.

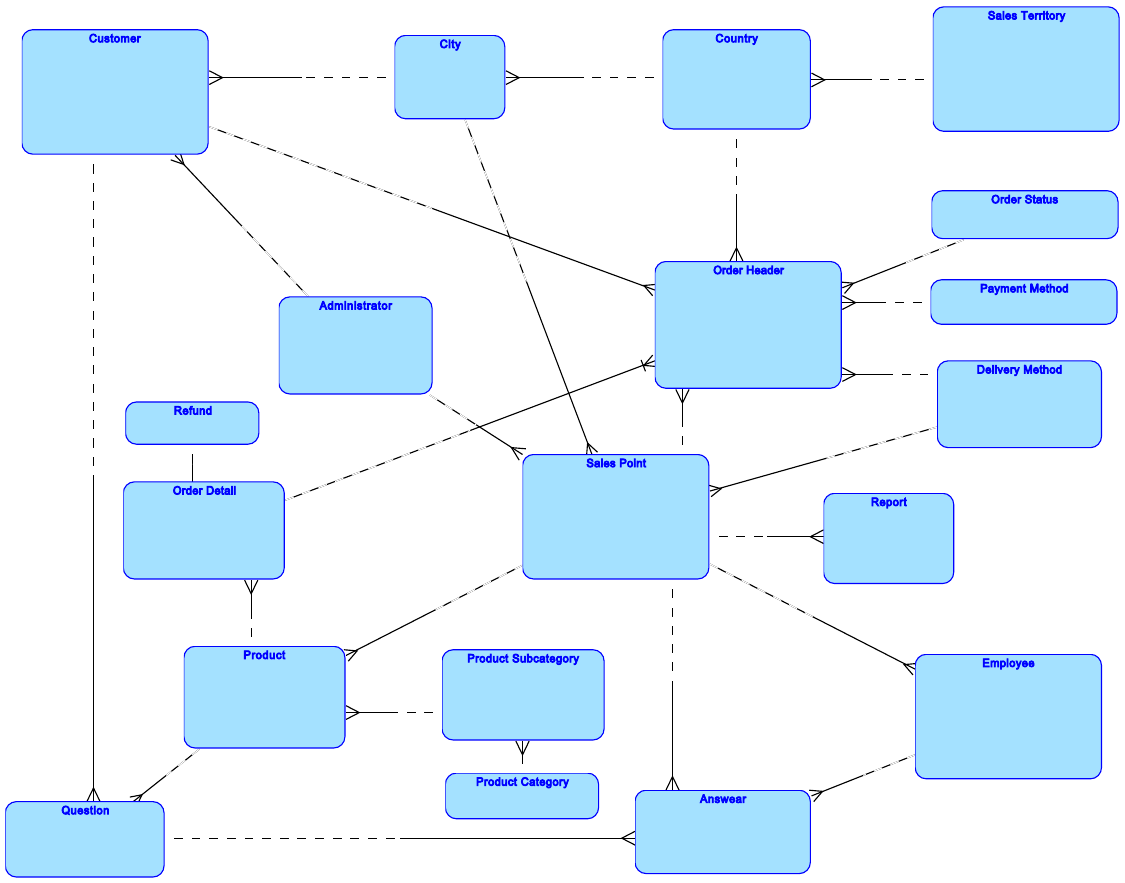
Obiekty rzeczywistości: pracownik, administrator, pytanie.

# Rozdział 4. Model konceptualny projektowanej bazy danych

Przedstaw w formie graficznej model konceptualny projektowanej bazy danych.

Wymień i scharakteryzuj encje, które będą uwzględnione w modelowanej bazie danych. Podaj ich nazwy oraz opisz ogólnie, jakie obiekty rzeczywistości będą one opisywały.

Opisz rodzaje związków pomiędzy poszczególnymi encjami.



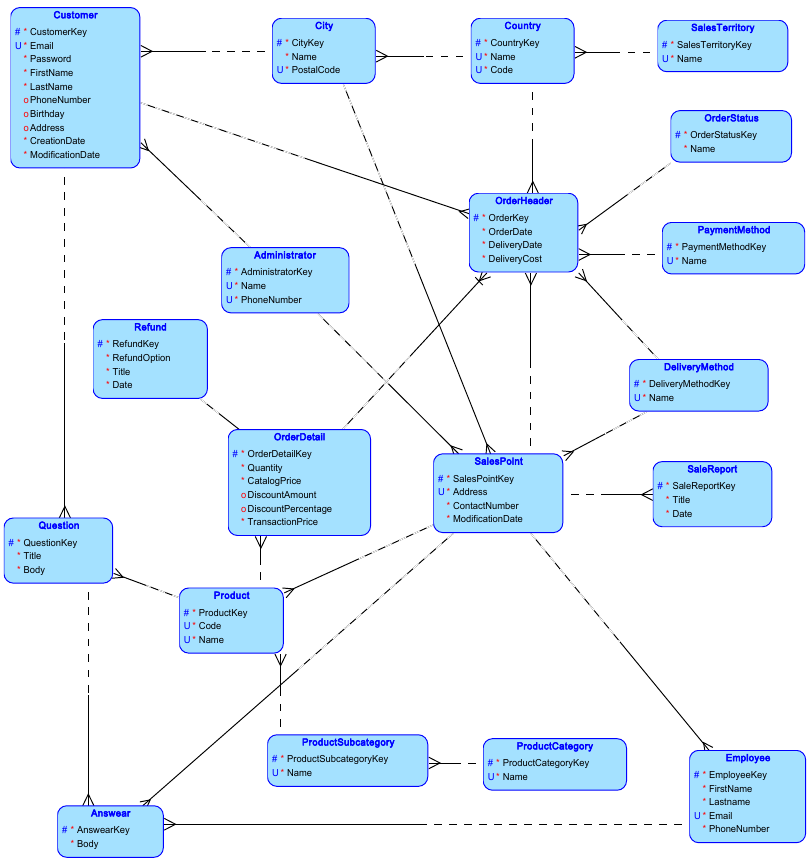
# Rozdział 5. Model związków encji projektowanej bazy danych

Przedstaw w formie graficznej model logiczny (związków encji) projektowanej bazy danych.

Wymień i opisz znaczenie atrybutów dla poszczególnych encji występujących w prezentowanym modelu logicznym bazy danych. Podaj nazwę i rodzaj każdego atrybutu encji.

Określ dla każdego atrybutu encji jego dziedzinę, maksymalny rozmiar danych, opcjonalność lub konieczność wystąpienia wartości określonej oraz inne ograniczenia integralnościowe, np. unikalność wartości atrybutu bądź też zawężenie jego dziedziny.

Dokonaj szczegółowego opisu każdego związku pomiędzy poszczególnymi encjami.

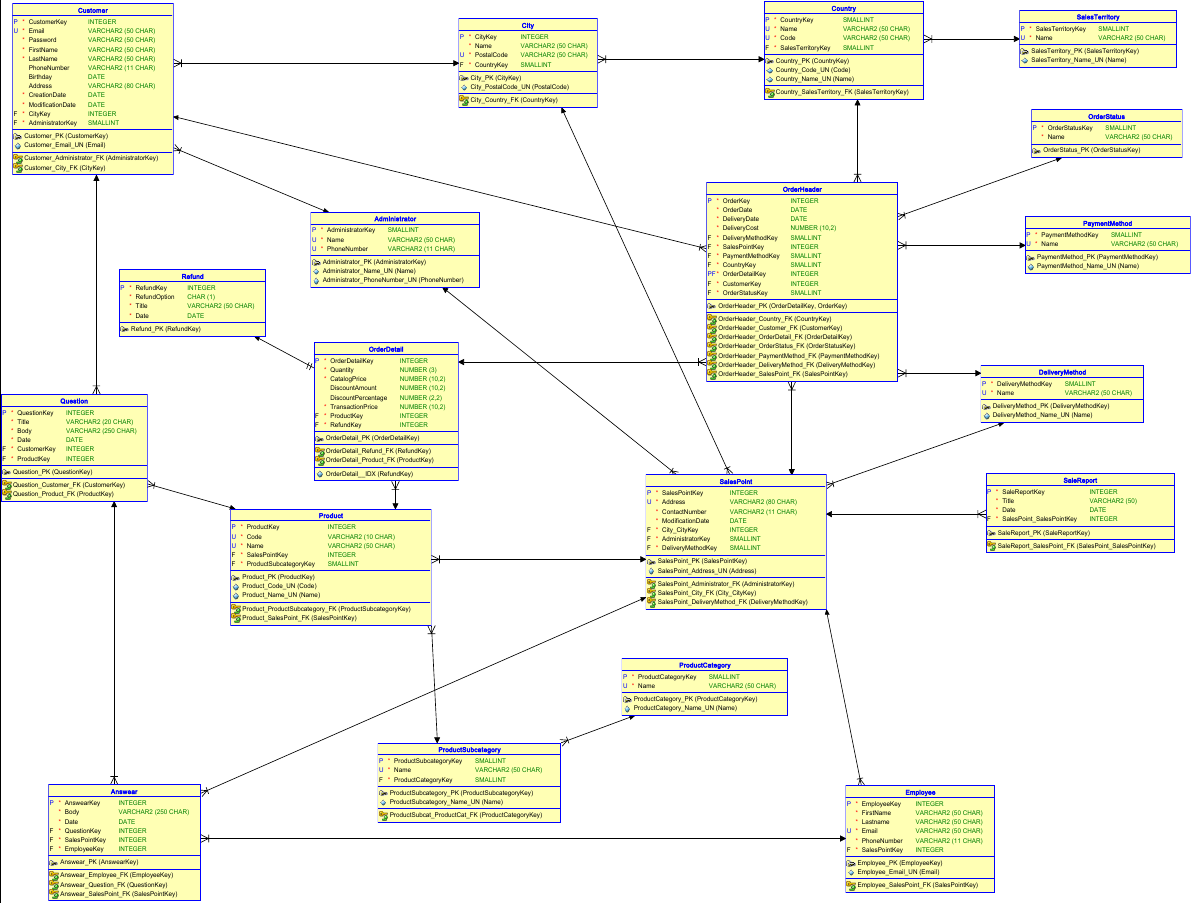


# Rozdział 6. Model relacyjny projektowanej bazy danych

Przedstaw w formie graficznej model relacyjny projektowanej bazy danych.

Określ ograniczenia integralnościowe zastosowane dla poszczególnych kolumn poszczególnych tabeli. Podaj nazwy tych ograniczeń oraz ich rodzaj.

Wymień indeksy, jakie należy utworzyć dla poszczególnych tabel. Podaj nazwę indeksu, jego rodzaj oraz kolumny użyte do jego zbudowania.



# Rozdział 7. Kod SQL – tworzenie bazy danych

Przedstaw instrukcje do tworzenia poszczególnych tabel bazy danych.

Przedstaw instrukcje do tworzenia więzów integralności zarówno na poziomie poszczególnych kolumn jak i całych tabel.

Przedstaw instrukcje do tworzenia związków między tabelami.

Przedstaw instrukcje do tworzenia indeksów dla poszczególnych tabel.

CREATE TABLE administrator (

administratorkey SMALLINT NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL,

phonenumber VARCHAR2(11 CHAR) NOT NULL

);

ALTER TABLE administrator ADD CONSTRAINT administrator\_pk PRIMARY KEY ( administratorkey );

ALTER TABLE administrator ADD CONSTRAINT administrator\_name\_un UNIQUE ( name );

ALTER TABLE administrator ADD CONSTRAINT administrator\_phonenumber\_un UNIQUE ( phonenumber );

CREATE TABLE answear (

answearkey INTEGER NOT NULL,

body VARCHAR2(250 CHAR) NOT NULL,

"Date" DATE NOT NULL,

questionkey INTEGER NOT NULL,

salespointkey INTEGER NOT NULL,

employeekey INTEGER NOT NULL

);

ALTER TABLE answear ADD CONSTRAINT answear\_pk PRIMARY KEY ( answearkey );

CREATE TABLE city (

citykey INTEGER NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL,

postalcode VARCHAR2(50 CHAR) NOT NULL,

countrykey SMALLINT NOT NULL

);

ALTER TABLE city ADD CONSTRAINT city\_pk PRIMARY KEY ( citykey );

ALTER TABLE city ADD CONSTRAINT city\_postalcode\_un UNIQUE ( postalcode );

CREATE TABLE country (

countrykey SMALLINT NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL,

code VARCHAR2(50 CHAR) NOT NULL,

salesterritorykey SMALLINT NOT NULL

);

ALTER TABLE country ADD CONSTRAINT country\_pk PRIMARY KEY ( countrykey );

ALTER TABLE country ADD CONSTRAINT country\_code\_un UNIQUE ( code );

ALTER TABLE country ADD CONSTRAINT country\_name\_un UNIQUE ( name );

CREATE TABLE customer (

customerkey INTEGER NOT NULL,

email VARCHAR2(50 CHAR) NOT NULL,

password VARCHAR2(50 CHAR) NOT NULL,

firstname VARCHAR2(50 CHAR) NOT NULL,

lastname VARCHAR2(50 CHAR) NOT NULL,

phonenumber VARCHAR2(11 CHAR),

birthday DATE,

address VARCHAR2(80 CHAR),

creationdate DATE NOT NULL,

modificationdate DATE NOT NULL,

citykey INTEGER NOT NULL,

administratorkey SMALLINT NOT NULL

);

ALTER TABLE customer ADD CONSTRAINT customer\_pk PRIMARY KEY ( customerkey );

ALTER TABLE customer ADD CONSTRAINT customer\_email\_un UNIQUE ( email );

CREATE TABLE deliverymethod (

deliverymethodkey SMALLINT NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL

);

ALTER TABLE deliverymethod ADD CONSTRAINT deliverymethod\_pk PRIMARY KEY ( deliverymethodkey );

ALTER TABLE deliverymethod ADD CONSTRAINT deliverymethod\_name\_un UNIQUE ( name );

CREATE TABLE employee (

employeekey INTEGER NOT NULL,

firstname VARCHAR2(50 CHAR) NOT NULL,

lastname VARCHAR2(50 CHAR) NOT NULL,

email VARCHAR2(50 CHAR) NOT NULL,

phonenumber VARCHAR2(11 CHAR) NOT NULL,

salespointkey INTEGER NOT NULL

);

ALTER TABLE employee ADD CONSTRAINT employee\_pk PRIMARY KEY ( employeekey );

ALTER TABLE employee ADD CONSTRAINT employee\_email\_un UNIQUE ( email );

CREATE TABLE orderdetail (

orderdetailkey INTEGER NOT NULL,

quantity NUMBER(3) NOT NULL,

catalogprice NUMBER(10, 2) NOT NULL,

discountamount NUMBER(10, 2),

discountpercentage NUMBER(2, 2),

transactionprice NUMBER(10, 2) NOT NULL,

productkey INTEGER NOT NULL,

refundkey INTEGER NOT NULL

);

CREATE UNIQUE INDEX orderdetail\_\_idx ON

orderdetail (

refundkey

ASC );

ALTER TABLE orderdetail ADD CONSTRAINT orderdetail\_pk PRIMARY KEY ( orderdetailkey );

CREATE TABLE orderheader (

orderkey INTEGER NOT NULL,

orderdate DATE NOT NULL,

deliverydate DATE NOT NULL,

deliverycost NUMBER(10, 2) NOT NULL,

deliverymethodkey SMALLINT NOT NULL,

salespointkey INTEGER NOT NULL,

paymentmethodkey SMALLINT NOT NULL,

countrykey SMALLINT NOT NULL,

orderdetailkey INTEGER NOT NULL,

customerkey INTEGER NOT NULL,

orderstatuskey SMALLINT NOT NULL

);

ALTER TABLE orderheader ADD CONSTRAINT orderheader\_pk PRIMARY KEY ( orderdetailkey,

orderkey );

CREATE TABLE orderstatus (

orderstatuskey SMALLINT NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL

);

ALTER TABLE orderstatus ADD CONSTRAINT orderstatus\_pk PRIMARY KEY ( orderstatuskey );

CREATE TABLE paymentmethod (

paymentmethodkey SMALLINT NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL

);

ALTER TABLE paymentmethod ADD CONSTRAINT paymentmethod\_pk PRIMARY KEY ( paymentmethodkey );

ALTER TABLE paymentmethod ADD CONSTRAINT paymentmethod\_name\_un UNIQUE ( name );

CREATE TABLE product (

productkey INTEGER NOT NULL,

code VARCHAR2(10 CHAR) NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL,

salespointkey INTEGER NOT NULL,

productsubcategorykey SMALLINT NOT NULL

);

ALTER TABLE product ADD CONSTRAINT product\_pk PRIMARY KEY ( productkey );

ALTER TABLE product ADD CONSTRAINT product\_code\_un UNIQUE ( code );

ALTER TABLE product ADD CONSTRAINT product\_name\_un UNIQUE ( name );

CREATE TABLE productcategory (

productcategorykey SMALLINT NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL

);

ALTER TABLE productcategory ADD CONSTRAINT productcategory\_pk PRIMARY KEY ( productcategorykey );

ALTER TABLE productcategory ADD CONSTRAINT productcategory\_name\_un UNIQUE ( name );

CREATE TABLE productsubcategory (

productsubcategorykey SMALLINT NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL,

productcategorykey SMALLINT NOT NULL

);

ALTER TABLE productsubcategory ADD CONSTRAINT productsubcategory\_pk PRIMARY KEY ( productsubcategorykey );

ALTER TABLE productsubcategory ADD CONSTRAINT productsubcategory\_name\_un UNIQUE ( name );

CREATE TABLE question (

questionkey INTEGER NOT NULL,

title VARCHAR2(20 CHAR) NOT NULL,

body VARCHAR2(250 CHAR) NOT NULL,

"Date" DATE NOT NULL,

customerkey INTEGER NOT NULL,

productkey INTEGER NOT NULL

);

ALTER TABLE question ADD CONSTRAINT question\_pk PRIMARY KEY ( questionkey );

CREATE TABLE refund (

refundkey INTEGER NOT NULL,

refundoption CHAR(1) NOT NULL,

title VARCHAR2(50 CHAR) NOT NULL,

"Date" DATE NOT NULL

);

ALTER TABLE refund ADD CONSTRAINT refund\_pk PRIMARY KEY ( refundkey );

CREATE TABLE salereport (

salereportkey INTEGER NOT NULL,

title VARCHAR2(50) NOT NULL,

"Date" DATE NOT NULL,

salespoint\_salespointkey INTEGER NOT NULL

);

ALTER TABLE salereport ADD CONSTRAINT salereport\_pk PRIMARY KEY ( salereportkey );

CREATE TABLE salespoint (

salespointkey INTEGER NOT NULL,

address VARCHAR2(80 CHAR) NOT NULL,

contactnumber VARCHAR2(11 CHAR) NOT NULL,

modificationdate DATE NOT NULL,

city\_citykey INTEGER NOT NULL,

administratorkey SMALLINT NOT NULL,

deliverymethodkey SMALLINT NOT NULL

);

ALTER TABLE salespoint ADD CONSTRAINT salespoint\_pk PRIMARY KEY ( salespointkey );

ALTER TABLE salespoint ADD CONSTRAINT salespoint\_address\_un UNIQUE ( address );

CREATE TABLE salesterritory (

salesterritorykey SMALLINT NOT NULL,

name VARCHAR2(50 CHAR) NOT NULL

);

ALTER TABLE salesterritory ADD CONSTRAINT salesterritory\_pk PRIMARY KEY ( salesterritorykey );

ALTER TABLE salesterritory ADD CONSTRAINT salesterritory\_name\_un UNIQUE ( name );

ALTER TABLE answear

ADD CONSTRAINT answear\_employee\_fk FOREIGN KEY ( employeekey )

REFERENCES employee ( employeekey );

ALTER TABLE answear

ADD CONSTRAINT answear\_question\_fk FOREIGN KEY ( questionkey )

REFERENCES question ( questionkey );

ALTER TABLE answear

ADD CONSTRAINT answear\_salespoint\_fk FOREIGN KEY ( salespointkey )

REFERENCES salespoint ( salespointkey );

ALTER TABLE city

ADD CONSTRAINT city\_country\_fk FOREIGN KEY ( countrykey )

REFERENCES country ( countrykey );

ALTER TABLE country

ADD CONSTRAINT country\_salesterritory\_fk FOREIGN KEY ( salesterritorykey )

REFERENCES salesterritory ( salesterritorykey );

ALTER TABLE customer

ADD CONSTRAINT customer\_administrator\_fk FOREIGN KEY ( administratorkey )

REFERENCES administrator ( administratorkey );

ALTER TABLE customer

ADD CONSTRAINT customer\_city\_fk FOREIGN KEY ( citykey )

REFERENCES city ( citykey );

ALTER TABLE employee

ADD CONSTRAINT employee\_salespoint\_fk FOREIGN KEY ( salespointkey )

REFERENCES salespoint ( salespointkey );

ALTER TABLE orderdetail

ADD CONSTRAINT orderdetail\_product\_fk FOREIGN KEY ( productkey )

REFERENCES product ( productkey );

ALTER TABLE orderdetail

ADD CONSTRAINT orderdetail\_refund\_fk FOREIGN KEY ( refundkey )

REFERENCES refund ( refundkey );

ALTER TABLE orderheader

ADD CONSTRAINT orderheader\_country\_fk FOREIGN KEY ( countrykey )

REFERENCES country ( countrykey );

ALTER TABLE orderheader

ADD CONSTRAINT orderheader\_customer\_fk FOREIGN KEY ( customerkey )

REFERENCES customer ( customerkey );

ALTER TABLE orderheader

ADD CONSTRAINT orderheader\_deliverymethod\_fk FOREIGN KEY ( deliverymethodkey )

REFERENCES deliverymethod ( deliverymethodkey );

ALTER TABLE orderheader

ADD CONSTRAINT orderheader\_orderdetail\_fk FOREIGN KEY ( orderdetailkey )

REFERENCES orderdetail ( orderdetailkey );

ALTER TABLE orderheader

ADD CONSTRAINT orderheader\_orderstatus\_fk FOREIGN KEY ( orderstatuskey )

REFERENCES orderstatus ( orderstatuskey );

ALTER TABLE orderheader

ADD CONSTRAINT orderheader\_paymentmethod\_fk FOREIGN KEY ( paymentmethodkey )

REFERENCES paymentmethod ( paymentmethodkey );

ALTER TABLE orderheader

ADD CONSTRAINT orderheader\_salespoint\_fk FOREIGN KEY ( salespointkey )

REFERENCES salespoint ( salespointkey );

ALTER TABLE product

ADD CONSTRAINT product\_productsubcategory\_fk FOREIGN KEY ( productsubcategorykey )

REFERENCES productsubcategory ( productsubcategorykey );

ALTER TABLE product

ADD CONSTRAINT product\_salespoint\_fk FOREIGN KEY ( salespointkey )

REFERENCES salespoint ( salespointkey );

ALTER TABLE productsubcategory

ADD CONSTRAINT productsubcat\_productcat\_fk FOREIGN KEY ( productcategorykey )

REFERENCES productcategory ( productcategorykey );

ALTER TABLE question

ADD CONSTRAINT question\_customer\_fk FOREIGN KEY ( customerkey )

REFERENCES customer ( customerkey );

ALTER TABLE question

ADD CONSTRAINT question\_product\_fk FOREIGN KEY ( productkey )

REFERENCES product ( productkey );

ALTER TABLE salereport

ADD CONSTRAINT salereport\_salespoint\_fk FOREIGN KEY ( salespoint\_salespointkey )

REFERENCES salespoint ( salespointkey );

ALTER TABLE salespoint

ADD CONSTRAINT salespoint\_administrator\_fk FOREIGN KEY ( administratorkey )

REFERENCES administrator ( administratorkey );

ALTER TABLE salespoint

ADD CONSTRAINT salespoint\_city\_fk FOREIGN KEY ( city\_citykey )

REFERENCES city ( citykey );

ALTER TABLE salespoint

ADD CONSTRAINT salespoint\_deliverymethod\_fk FOREIGN KEY ( deliverymethodkey )

REFERENCES deliverymethod ( deliverymethodkey );

CREATE SEQUENCE administrator\_administratorkey START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER administrator\_administratorkey BEFORE

INSERT ON administrator

FOR EACH ROW

WHEN ( new.administratorkey IS NULL )

BEGIN

:new.administratorkey := administrator\_administratorkey.nextval;

END;

/

CREATE SEQUENCE answear\_answearkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER answear\_answearkey\_trg BEFORE

INSERT ON answear

FOR EACH ROW

WHEN ( new.answearkey IS NULL )

BEGIN

:new.answearkey := answear\_answearkey\_seq.nextval;

END;

/

CREATE SEQUENCE city\_citykey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER city\_citykey\_trg BEFORE

INSERT ON city

FOR EACH ROW

WHEN ( new.citykey IS NULL )

BEGIN

:new.citykey := city\_citykey\_seq.nextval;

END;

/

CREATE SEQUENCE country\_countrykey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER country\_countrykey\_trg BEFORE

INSERT ON country

FOR EACH ROW

WHEN ( new.countrykey IS NULL )

BEGIN

:new.countrykey := country\_countrykey\_seq.nextval;

END;

/

CREATE SEQUENCE customer\_customerkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER customer\_customerkey\_trg BEFORE

INSERT ON customer

FOR EACH ROW

WHEN ( new.customerkey IS NULL )

BEGIN

:new.customerkey := customer\_customerkey\_seq.nextval;

END;

/

CREATE SEQUENCE employee\_employeekey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER employee\_employeekey\_trg BEFORE

INSERT ON employee

FOR EACH ROW

WHEN ( new.employeekey IS NULL )

BEGIN

:new.employeekey := employee\_employeekey\_seq.nextval;

END;

/

CREATE SEQUENCE orderdetail\_orderdetailkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER orderdetail\_orderdetailkey\_trg BEFORE

INSERT ON orderdetail

FOR EACH ROW

WHEN ( new.orderdetailkey IS NULL )

BEGIN

:new.orderdetailkey := orderdetail\_orderdetailkey\_seq.nextval;

END;

/

CREATE SEQUENCE orderheader\_orderkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER orderheader\_orderkey\_trg BEFORE

INSERT ON orderheader

FOR EACH ROW

WHEN ( new.orderkey IS NULL )

BEGIN

:new.orderkey := orderheader\_orderkey\_seq.nextval;

END;

/

CREATE SEQUENCE orderstatus\_orderstatuskey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER orderstatus\_orderstatuskey\_trg BEFORE

INSERT ON orderstatus

FOR EACH ROW

WHEN ( new.orderstatuskey IS NULL )

BEGIN

:new.orderstatuskey := orderstatus\_orderstatuskey\_seq.nextval;

END;

/

CREATE SEQUENCE paymentmethod\_paymentmethodkey START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER paymentmethod\_paymentmethodkey BEFORE

INSERT ON paymentmethod

FOR EACH ROW

WHEN ( new.paymentmethodkey IS NULL )

BEGIN

:new.paymentmethodkey := paymentmethod\_paymentmethodkey.nextval;

END;

/

CREATE SEQUENCE product\_productkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER product\_productkey\_trg BEFORE

INSERT ON product

FOR EACH ROW

WHEN ( new.productkey IS NULL )

BEGIN

:new.productkey := product\_productkey\_seq.nextval;

END;

/

CREATE SEQUENCE productcategory\_productcategor START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER productcategory\_productcategor BEFORE

INSERT ON productcategory

FOR EACH ROW

WHEN ( new.productcategorykey IS NULL )

BEGIN

:new.productcategorykey := productcategory\_productcategor.nextval;

END;

/

CREATE SEQUENCE productsubcategory\_productsubc START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER productsubcategory\_productsubc BEFORE

INSERT ON productsubcategory

FOR EACH ROW

WHEN ( new.productsubcategorykey IS NULL )

BEGIN

:new.productsubcategorykey := productsubcategory\_productsubc.nextval;

END;

/

CREATE SEQUENCE question\_questionkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER question\_questionkey\_trg BEFORE

INSERT ON question

FOR EACH ROW

WHEN ( new.questionkey IS NULL )

BEGIN

:new.questionkey := question\_questionkey\_seq.nextval;

END;

/

CREATE SEQUENCE refund\_refundkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER refund\_refundkey\_trg BEFORE

INSERT ON refund

FOR EACH ROW

WHEN ( new.refundkey IS NULL )

BEGIN

:new.refundkey := refund\_refundkey\_seq.nextval;

END;

/

CREATE SEQUENCE salereport\_salereportkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER salereport\_salereportkey\_trg BEFORE

INSERT ON salereport

FOR EACH ROW

WHEN ( new.salereportkey IS NULL )

BEGIN

:new.salereportkey := salereport\_salereportkey\_seq.nextval;

END;

/

CREATE SEQUENCE salespoint\_salespointkey\_seq START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER salespoint\_salespointkey\_trg BEFORE

INSERT ON salespoint

FOR EACH ROW

WHEN ( new.salespointkey IS NULL )

BEGIN

:new.salespointkey := salespoint\_salespointkey\_seq.nextval;

END;

/

CREATE SEQUENCE salesterritory\_salesterritoryk START WITH 1 NOCACHE ORDER;

CREATE OR REPLACE TRIGGER salesterritory\_salesterritoryk BEFORE

INSERT ON salesterritory

FOR EACH ROW

WHEN ( new.salesterritorykey IS NULL )

BEGIN

:new.salesterritorykey := salesterritory\_salesterritoryk.nextval;

END;

/





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