MINIPROJEKT

DOKUMENTACJA

Krystian Walat, Marcin Pałczak, Wojciech Piechaczek

1. Opis projektu

Celem projektu jest stworzenie aplikacji desktopowej dla biura turystycznego.
Technologie:

- korzystamy z bazy Oracle udostępnianej przez AGH
- aplikacja kliencka napisana jest w Javie z wykorzystaniem frameworka Hibernate

Zrealizowane funkcjonalności:

- rejestracja, logowanie oraz system ról
- przeglądanie wycieczek przez użytkowników
- rezerwowanie wycieczek przez użytkowników
- ocenianie wycieczek przez użytkowników

2. Wymagania funkcjonalne

a) rejestracja:
   - użytkownik rejestruje się,podając login i hasło oraz dane osobowe: imię, nazwisko, numer telefonu i adres e-mail
   - użytkownik wybiera również swoją rolę
   - login jest unikatowy i nie może się powtarzać
   - numer telefonu musi zawierać 9 cyfr
   - adres e-mail musi być poprawny, np. abc@example.com
   - jeśli użytkownik poda niepoprawne dane, aplikacja zwróci informację o tym, które są błędne
   - hasło przed zapisaniem jest haszowane i nie jest przechowywane w jawnej postaci

b) logowanie:
   - użytkownik loguje się, podając zarejestrowany login i hasło
   - jeżeli login jest niepoprawny, użytkownik otrzymuje informację o braku takiego konta w bazie
   - jeżeli hasło jest niepoprawne, użytkownik otrzymuje informację o niezgodności danych

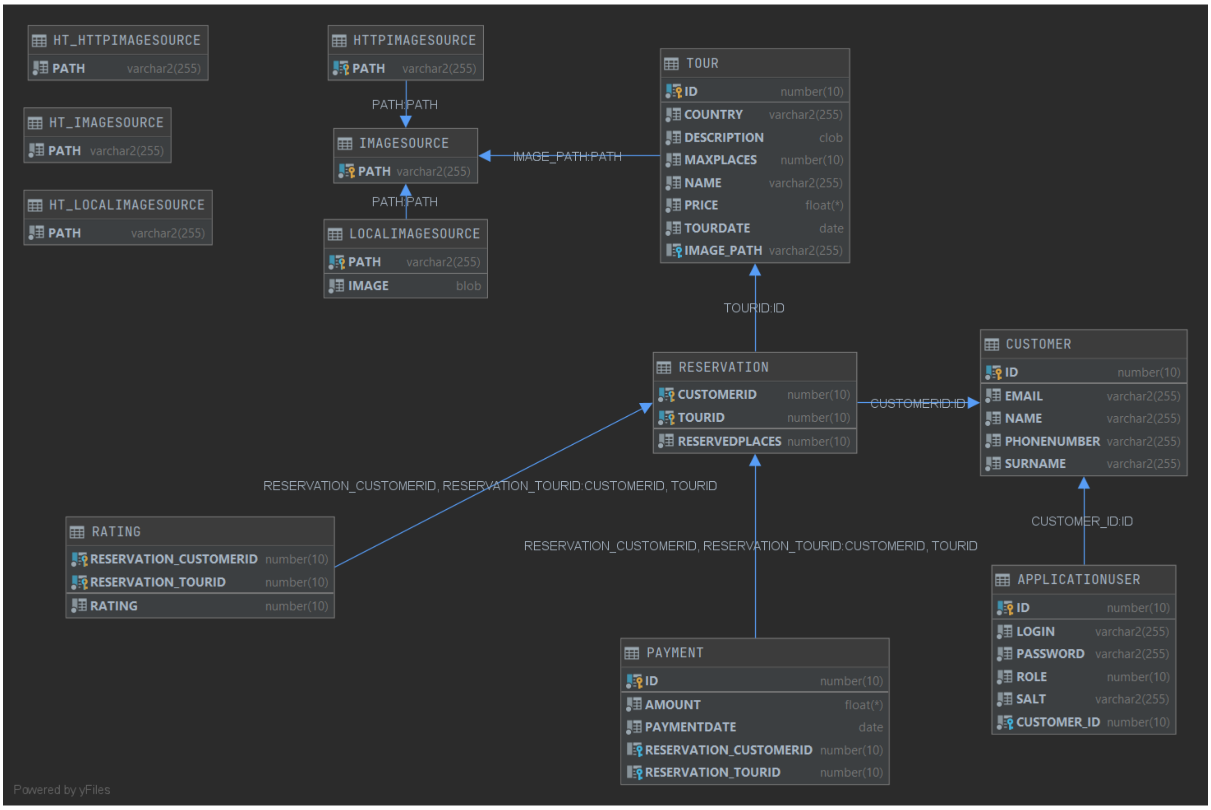
c) przeglądanie listy wycieczek:
   - po pomyślnym zalogowaniu, użytkownik może przeglądać listę wycieczek
   - wycieczki pojawiają się w formie kolejnych bloków na ekranie
   - każda wycieczka wyświetla podstawowe informacje o sobie: nazwa, kraj, cena jednostkowa, data oraz ilość aktualnie wolnych miejsc i maksymalna ilość miejsc
   - użytkownik widzi swoją nazwę i numer

- użytkownik może się również wylogować, co przenosi go z powrotem na ekran logowania
- użytkownik może również przejść w szczegóły wycieczki

d) szczegóły wycieczki:

- poza informacjami podanymi na liście wycieczek, szczegóły wycieczki zawierają również jej opis, ocenę oraz zdjęcie
- na tym panelu można zarezerwować wycieczkę oraz anulować rezerwację
- na rezerwacji znajduje się ilość zarezerwowanych miejsc
- każdy klient dla danej wycieczki może mieć tylko jedną rezerwację
- rezerwacji można dokonać tylko zanim wycieczka się rozpoczęła
- rezerwację można anulować tylko jeśli nie została już opłacona, lub jeśli wycieczka się jeszcze nie odbyła
- możliwe jest tutaj również wystawienie oceny wycieczki
- ocenić wycieczkę można tylko wtedy, gdy posiada się rezerwację na co najmniej jedno miejsce dla danej wycieczki
- anulowanie całej rezerwacji wiąże się z cofnięciem wystawionej oceny

3. Schemat bazy danych



Niepołączone z diagramem tabele z przedrostkiem HT_ są tabelami tymczasowymi generowanymi automatycznie przez Hibernate'a.

4. Przegląd ciekawych decyzji projektowych

4.1 Stosowanie funkcji zdefiniowanych w bazie danych zamiast niektórych zapytań

Zauważyliśmy, że wykonanie zapytań HQL postaci “FROM Table”, gdzie Table to tabela zawierająca klucze obce wymaga pobrania wszystkich informacji z tabeli połączonej kluczem obcym. Może to być nieefektywne, jeśli nie wymagamy pełnej informacji z tabeli, a interesują nas jedynie pojedyncze kolumny.

Dla szczególnie często używanych zapytań (średnia ocena, ilość ocen, ilość wolnych miejsc), stwierdziliśmy że zamiast wykonywać zapytania “SELECT x FROM y” zdefiniujemy te zapytania jako funkcje w bazie danych. Decyzja ta wynika z faktu, że wyniki tych funkcji są istotnymi informacjami, do których dostęp powinny mieć również inne aplikacje chcące korzystać z naszej bazy danych.

Aby wywołać funkcje korzystamy ze sztucznej tabeli “dual” dostępnej w bazie Oracle mającą tylko jedną kolumnę.

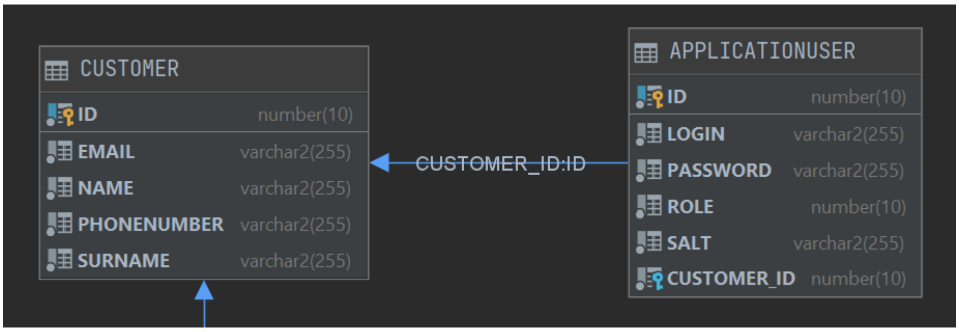
W kodzie wywołania odwołujące się do funkcji zdefiniowanych na bazie danych wyglądają następująco:

```
session
.createNativeQuery(
"select AVAILABLE_PLACES(:id) from dual")
.setParameter("id", this.id)
.getSingleResult();
```

Gdzie w powyższym przykładzie AVAILABLE_PLACES jest nazwą funkcji zdefiniowanej w bazie danych.

4.2 Rozdzielenie tabel Customer, oraz ApplicationUser

W bazie danych pojedynczego użytkownika opisują dwie tabele: Customer, oraz ApplicationUser, przy czym ApplicationUser zawiera klucz obcy odnoszący się do tabeli Customer.



CUSTOMER	
ID	number(10)
EMAIL	varchar2(255)
NAME	varchar2(255)
PHONENUMBER	varchar2(255)
SURNAME	varchar2(255)

APPLICATIONUSER	
ID	number(10)
LOGIN	varchar2(255)
PASSWORD	varchar2(255)
ROLE	number(10)
SALT	varchar2(255)
CUSTOMER_ID	number(10)

Podział ten wynika z faktu, że zaprojektowaliśmy bazę w taki sposób, aby umożliwić ewentualne jej rozszerzenie na więcej aplikacji. Tabela Customer zawiera informacje kontaktowe klienta i jest stosowana jako klucz obcy np.: dla rezerwacji.
Tabela ApplicationUser przechowuje informacje logowania użytkownika konkretnie dla naszej aplikacji.
Dzięki takiemu podziałowi dodanie np.: nowego sposobu autoryzacji nie będzie wymagało zmian w tabeli Customer.

4.3 Composite primary key w tabeli Reservation

W tabeli Reservation para ID klienta i ID wycieczki stanowi klucz główny.
Uzyskanie takiego efektu za pomocą Hibernate'a wymaga nieco wysiłku, szczególnie że obie kolumny będące kluczem głównym są równocześnie kluczami obcymi.

Rozwiązanie jakie wykorzystaliśmy to metoda EmbeddedID, polegająca na tym, że utworzona zostaje osobna klasa reprezentująca klucz, a następnie zostaje ona wbudowana do klasy Reservation. Następnie odpowiednie pola klasy Reservation są oznaczone jako klucze obce, wzięte z wbudowanej klasy.

Minimalny kod obrazujący to rozwiązanie:
Klasa reprezentująca samo ID

```
@Embeddable
public class ReservationID implements Serializable {
   private Integer customerID;
   private Integer tourID;

   public ReservationID() {
       //required by Hibernate
   }

   …

}
```

Klasa, w którą zostaje wbudowane złożone ID (@EmbeddedID)

```
@Entity
public class Reservation {

   @EmbeddedId
   private ReservationID reservationID;

   @MapsId("customerID")
   @ManyToOne(optional = false)
   @JoinColumns(value = {
           @JoinColumn(name = "customerID", referencedColumnName = "id")
})
   private Customer customer;

   @MapsId("tourID")
   @ManyToOne(optional = false)
   @JoinColumns(value = {
           @JoinColumn(name = "tourID", referencedColumnName = "id") })
   private Tour tour;

   private int reservedPlaces;

   public Reservation() {
       //required by Hibernate
       this.reservationID = new ReservationID();
   }

   …

}
```

4.4 Zastosowanie BLOB i CLOB w bazie danych

Aby przechowywać cache zdjęć w bazie danych zdecydowaliśmy się użyć typu BLOB w tabeli LocalImageSource.

Od strony kodu wymaga to dodania adnotacji @Lob (w przypadku zdjęcia zdefiniowaliśmy *explicite*, że chcemy mapować do typu BLOB). Jako typ zmiennej przechowującej zdjęcie ustawiliśmy tablicę bajtów, pozwalając tym samym bezproblemowy zapis zdjęcia w bazie danych.

```
@Entity
public class LocalImageSource extends ImageSource{

    @Lob
    @Column(nullable=false, columnDefinition="BLOB")
    protected byte[] image;

    public LocalImageSource() {
        //required by hibernate
    }

    …

}
```

Podobnie wykorzystaliśmy adnotację @Lob, aby przechować opis wycieczki jako CLOB. Wyszliśmy tutaj z założenia, że opis wycieczki w ogólności może być bardzo obszerny i stąd efektywnie będzie go przechowywać w taki sposób. Dodatkowo chcieliśmy się dowiedzieć w jaki sposób wygląda to od strony kodu.

Oznaczenie pola typu String adnotacją @Lob wystarcza, aby w bazie przyjęło ono typ CLOB

```
@Entity
public class Tour {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;

    …

    @Lob
    @Column(nullable = false)
    private String description;

    public Tour(){
        //required by Hibernate
    }

    …

}
```

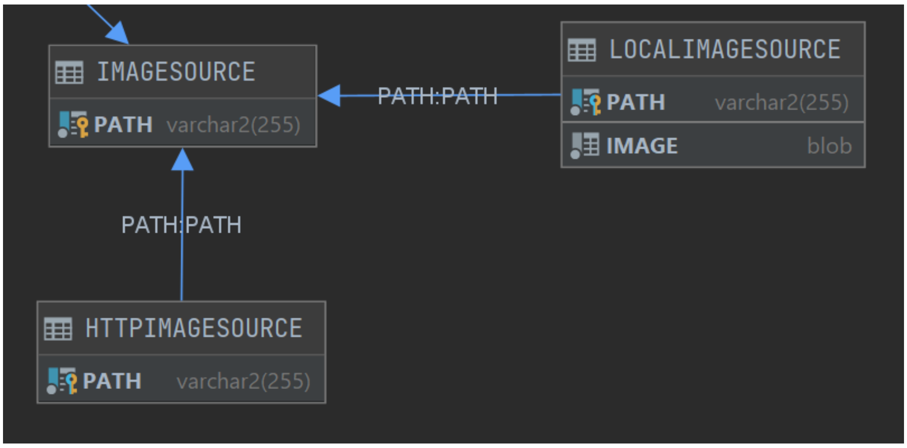
4.5 Dziedziczenie w tabeli ImageSource

Zdecydowaliśmy pozwalać w bazie danych na przechowywanie bezpośrednio zdjęć wycieczki jako BLOB, oraz na przechowywanie jedynie URL, z którego pobrane zostanie zdjęcie. Od strony kodu wygląda to tak, że zaimplementowaliśmy strukturę dziedziczenia gdzie klasą bazową jest ImageSource, a jej podklasami są HttpImageSource i LocalImageSource.

W teorii wystarczyłoby przechowywać w bazie sam URL jako String w jednej tabeli, oraz BLOB w drugiej. Jednak w takiej implementacji sprawdzanie czy zdjęcie zapisane jako BLOB już istnieje w bazie danych byłoby mocno nieefektywne, a chcemy uniknąć duplikatów zdjęć w bazie danych. W związku z tym myśleliśmy nad tym, aby dodać pewne unikalne id do tabeli przechowującej BLOBy ze zdjęciami. Jednak przy takim rozwiązaniu bardzo łatwo o błąd ludzki i dodanie wielokrotnie tego samego zdjęcia z różnymi ID.

W związku z tym zdecydowaliśmy, aby unikalnym identyfikatorem zdjęcia zapisanego jako BLOB była jego ścieżka w folderze /resources projektu. To rozwiązanie również ma swoje wady, np.: wymaga porównywania Stringów przy dostępie do tabeli, jednak pozwoliło mocno wyeliminować redundancję danych spowodowaną błędem ludzkim, więc zdecydowaliśmy się zachować to rozwiązanie. Dodatkowo takie rozwiązanie pozwoliło wydzielić String będący ścieżką zdjęcia do klasy nadrzędnej.

Mapując strukturę dziedziczenia do bazy danych wykorzystaliśmy strategię ”Joined”, w której klasy podrzędne zawierają jako klucz obcy wskazanie na tabelę nadrzędną.



4.6 Mapowanie Javowego Enuma do bazy danych

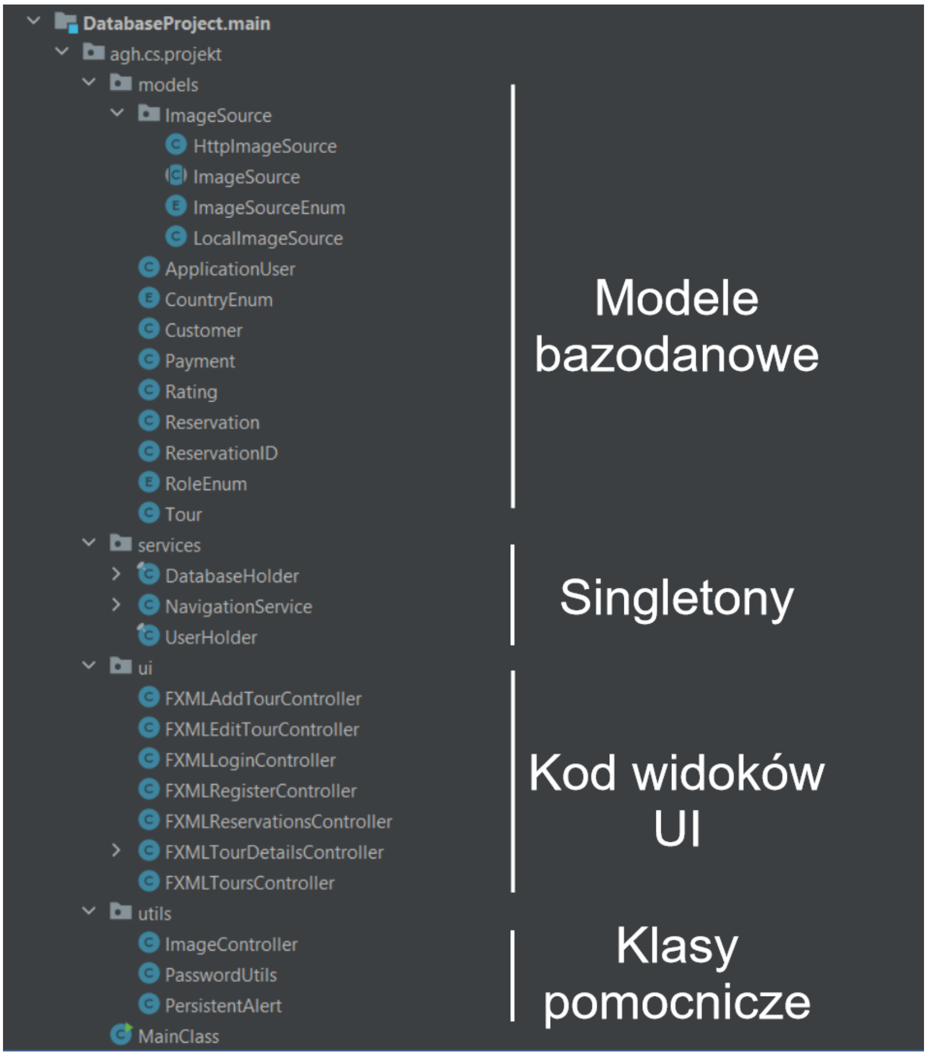
Zastanawialiśmy się w jaki sposób Enum w javie zostanie zmapowany do bazy danych. Kraj docelowy wycieczki w kodzie reprezentujemy jako Enum przyjmujący jedną z ustalonych wartości. Domyślny sposób mapowania spowodował, że w bazie danych przechowywane były jedynie indeksy poszczególnych wartości Enuma.
Jest to rozwiązanie bardzo wydajne, jednak zmniejsza czytelność bazy danych, ponieważ kraje są jako liczby. Tak samo inne aplikacje nie miałyby dostępu do tego, który numer odpowiada której nazwie kraju.

Hibernate pozwala jednak przechowywać Enumy po ich nazwie, zamiast indeksie przy zastosowaniu odpowiedniej adnotacji i jest to rozwiązanie, które ostatecznie wykorzystaliśmy.

```
@Entity
public class Tour {

   @Id
   @GeneratedValue(strategy = GenerationType.AUTO)
   private int id;

   @Enumerated(EnumType.STRING)
   @Column(nullable = false)
   private CountryEnum country;

   …

}
```

W ten sposób kraje przechowywane są czytelnie, po nazwie, a w kodzie wciąż możemy korzystać z nich jako Enum.

5. Struktura projektu

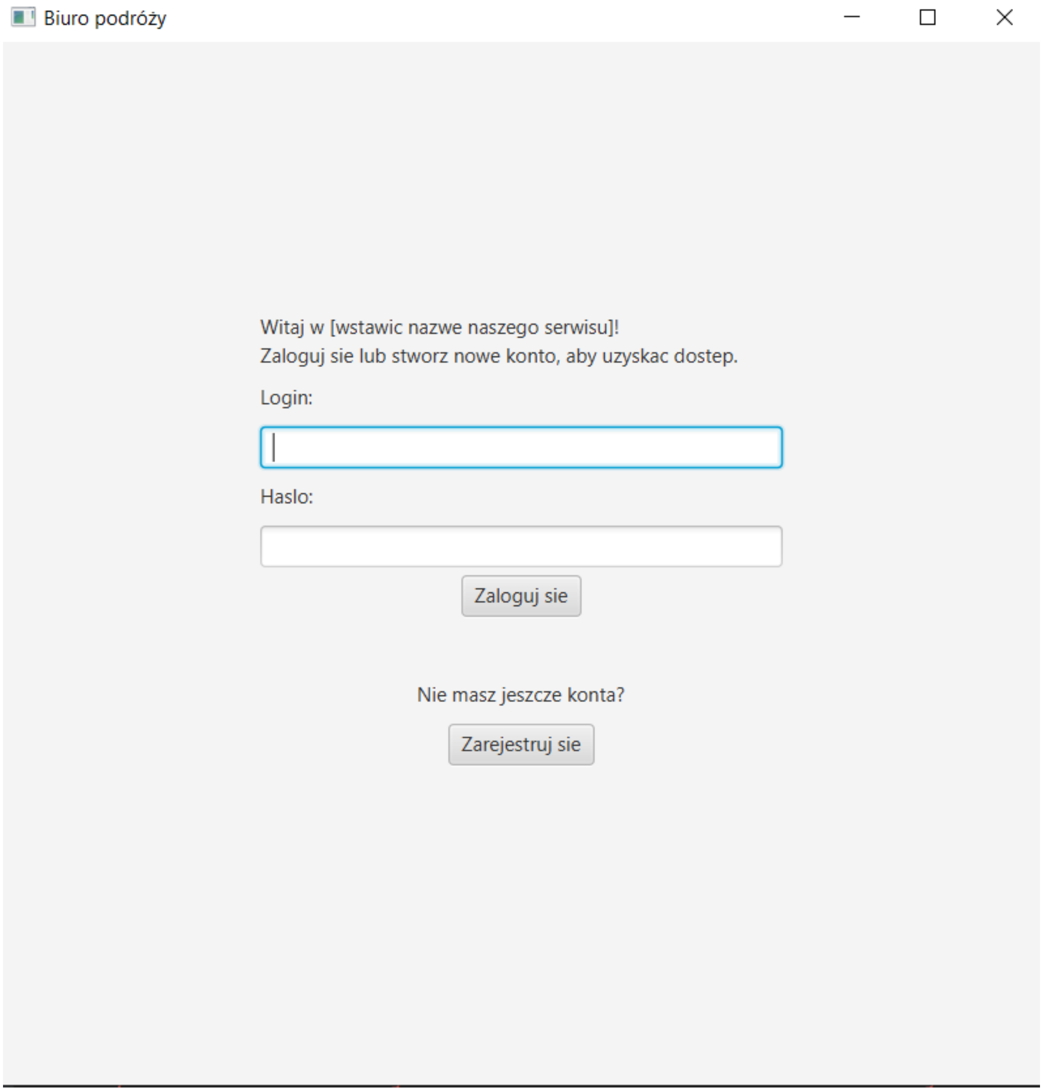


```
DatabaseProject.main
  agh.cs.projekt
    models
      ImageSource
        HttpImageSource
        ImageSource
        ImageSourceEnum
        LocalImageSource
      ApplicationUser
      CountryEnum
      Customer
      Payment
      Rating
      Reservation
      ReservationID
      RoleEnum
      Tour
    services
      DatabaseHolder
      NavigationService
      UserHolder
    ui
      FXMLAddTourController
      FXMLEditTourController
      FXMLLoginController
      FXMLRegisterController
      FXMLReservationsController
      FXMLTourDetailsController
      FXMLToursController
    utils
      ImageController
      PasswordUtils
      PersistentAlert
    MainClass
```

6. Warstwa wizualna

Pod spodem przedstawiamy zdjęcia wyglądu aplikacji, podając również nazwy klas odpowiadające w kodzie za wyświetlanie każdego z widoków.

Logowanie (FXMLLoginController)



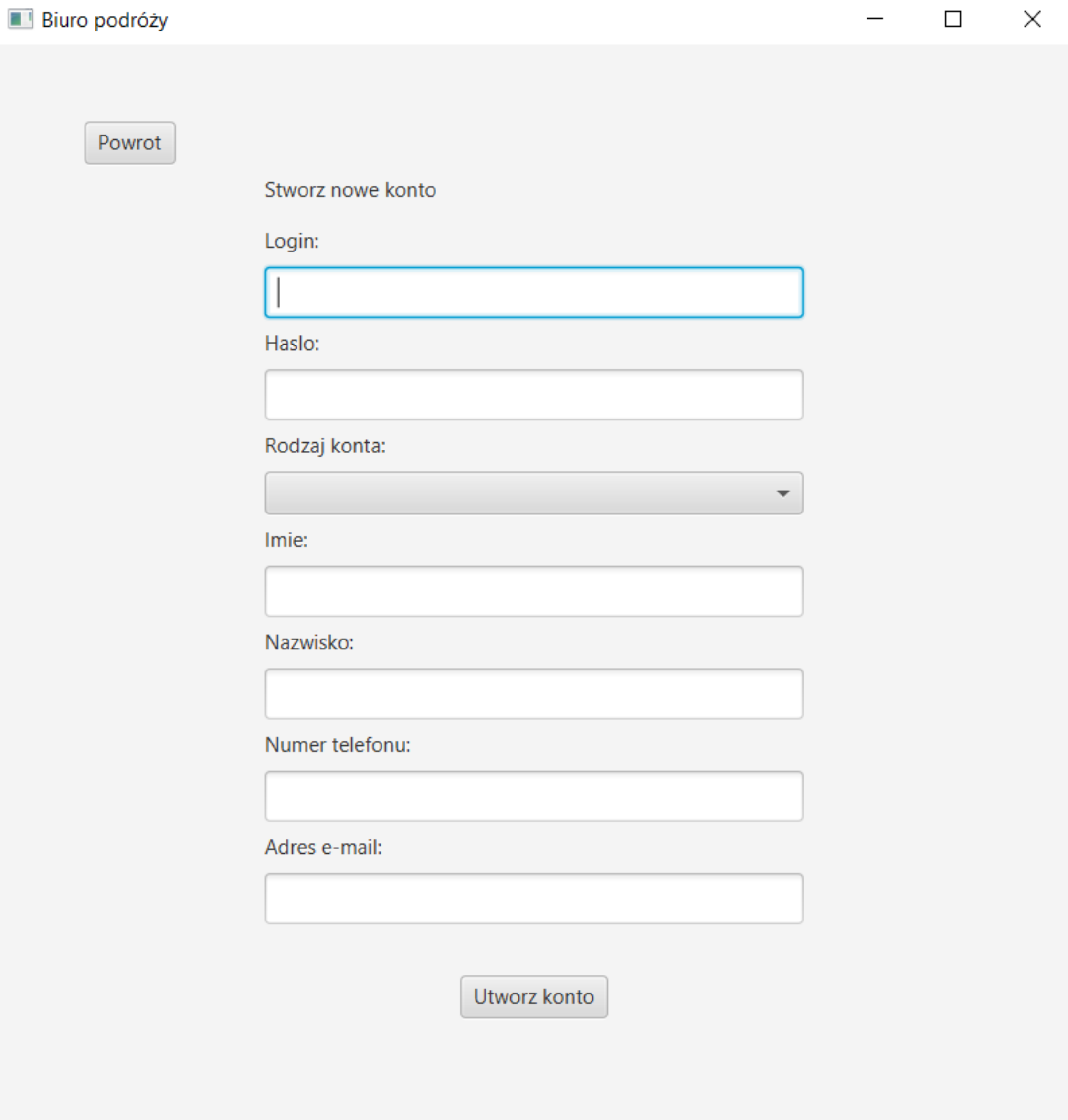
Witaj w [wstawic nazwe naszego serwisu]!
Zaloguj sie lub stworz nowe konto, aby uzyskac dostep.

Login:

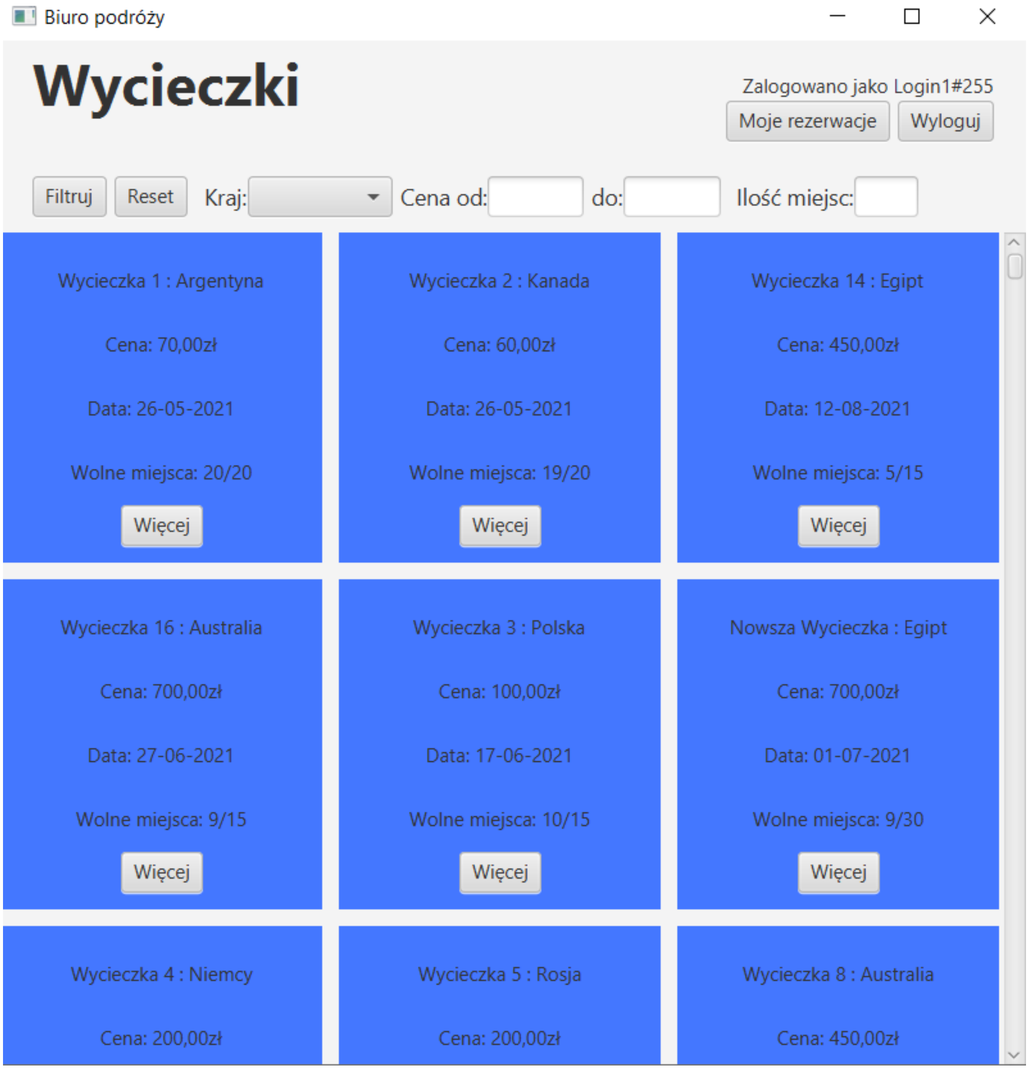
Haslo:

Nie masz jeszcze konta?

Rejestracja (FXMLRegisterController)



Lista wycieczek (FXMLToursController)



Biuro podróży

Wycieczki

Zalogowano jako Login1#255

Wycieczka 1 : Argentyna
Cena: 70,00zł
Data: 26-05-2021
Wolne miejsca: 20/20

Wycieczka 2 : Kanada
Cena: 60,00zł
Data: 26-05-2021
Wolne miejsca: 19/20

Wycieczka 14 : Egipt
Cena: 450,00zł
Data: 12-08-2021
Wolne miejsca: 5/15

Wycieczka 16 : Australia
Cena: 700,00zł
Data: 27-06-2021
Wolne miejsca: 9/15

Wycieczka 3 : Polska
Cena: 100,00zł
Data: 17-06-2021
Wolne miejsca: 10/15

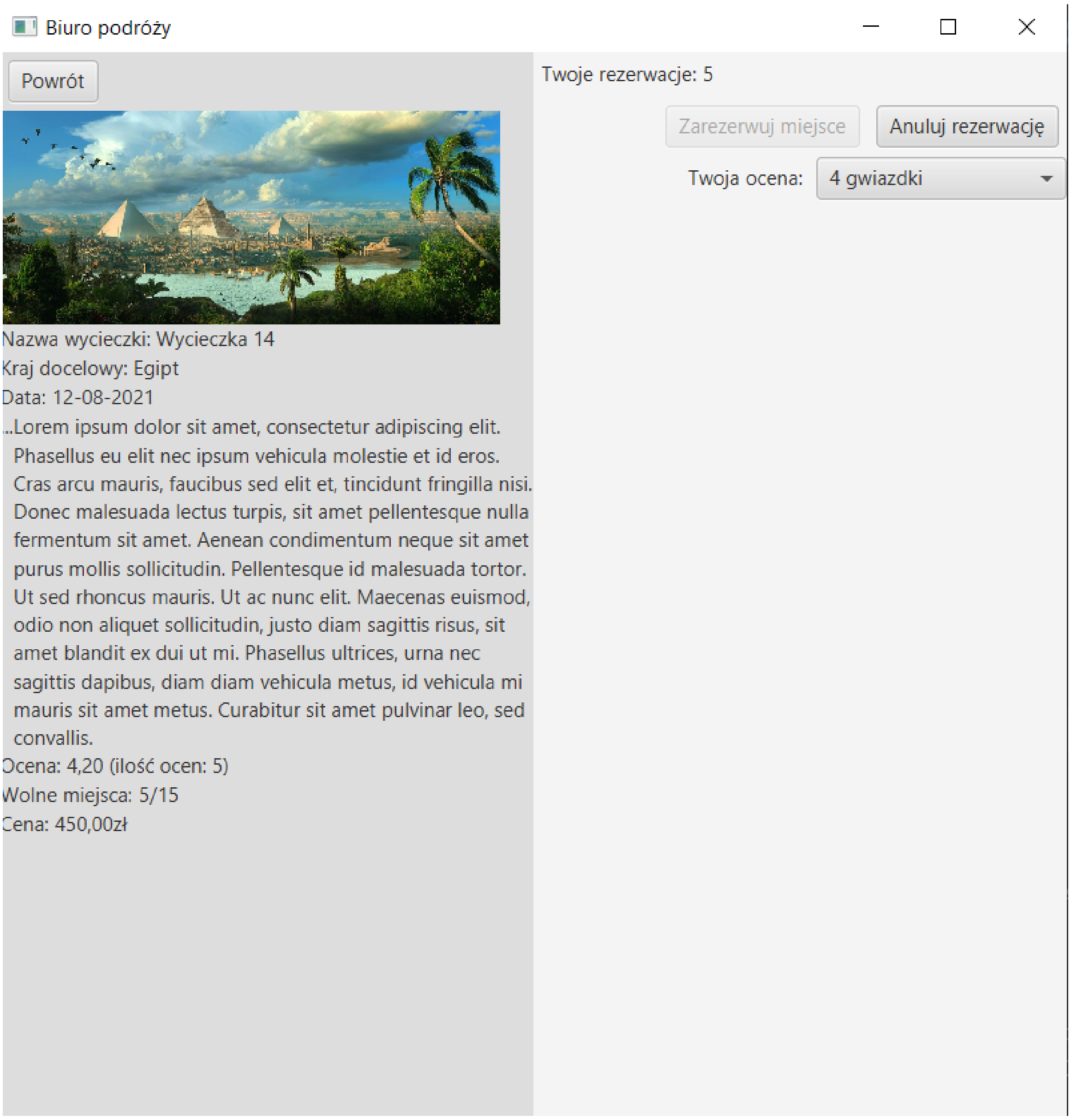
Nowsza Wycieczka : Egipt
Cena: 700,00zł
Data: 01-07-2021
Wolne miejsca: 9/30

Wycieczka 4 : Niemcy
Cena: 200,00zł

Wycieczka 5 : Rosja
Cena: 200,00zł

Wycieczka 8 : Australia
Cena: 450,00zł

Szczegóły wycieczki (FXMLTourDetailsController)



Twoje rezerwacje: 5

Twoja ocena: 4 gwiazdki

Nazwa wycieczki: Wycieczka 14

Kraj docelowy: Egipt

Data: 12-08-2021

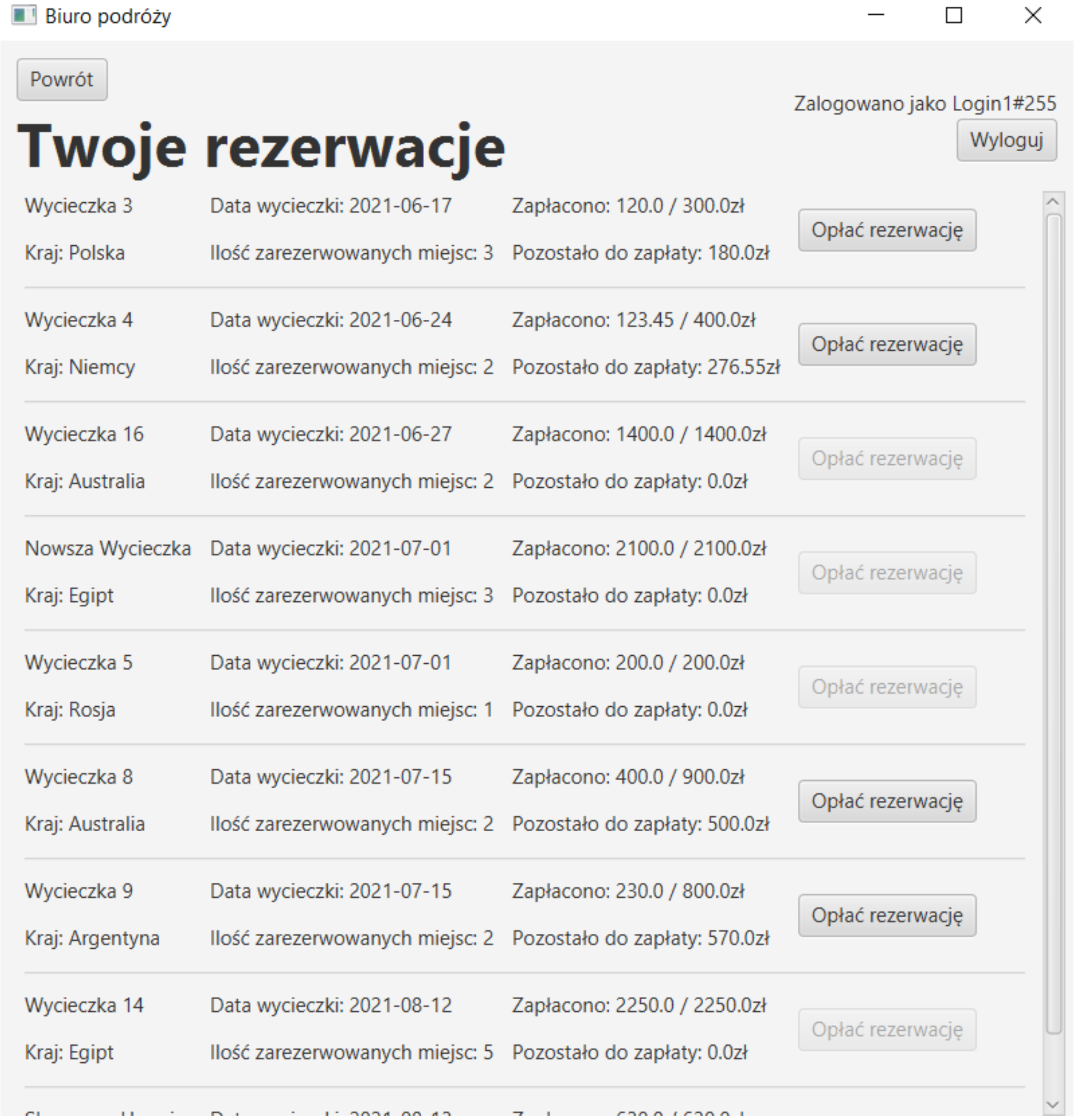
...Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi. Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed convallis.

Ocena: 4,20 (ilość ocen: 5)

Wolne miejsca: 5/15

Cena: 450,00zł

Koszyk (FXMLReservationsController)

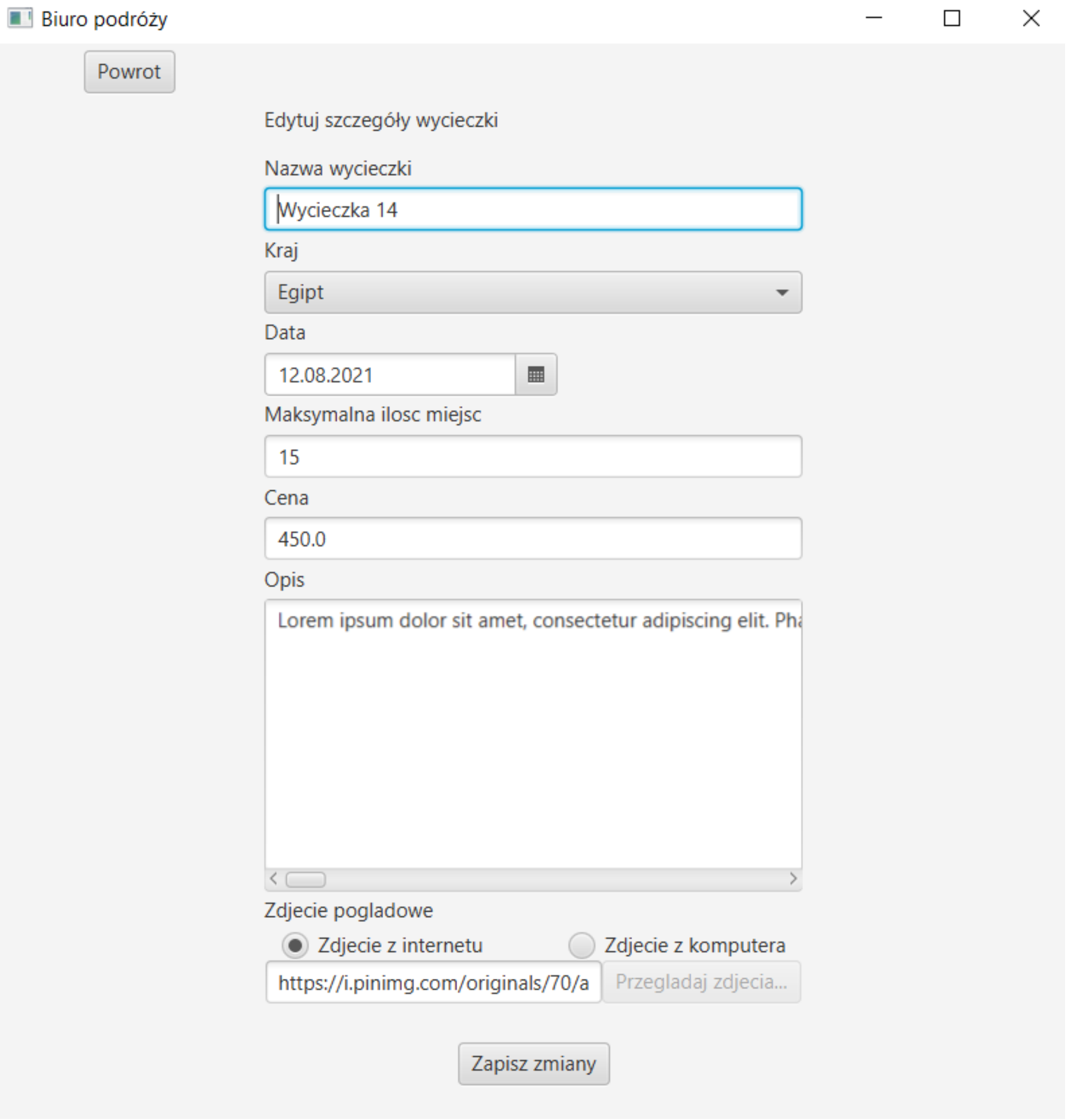


Zalogowano jako Login1#255

Twoje rezerwacje

Wycieczka 3 Kraj: Polska	Data wycieczki: 2021-06-17 Ilość zarezerwowanych miejsc: 3	Zapłacono: 120.0 / 300.0zł Pozostało do zapłaty: 180.0zł	Opłać rezerwację
Wycieczka 4 Kraj: Niemcy	Data wycieczki: 2021-06-24 Ilość zarezerwowanych miejsc: 2	Zapłacono: 123.45 / 400.0zł Pozostało do zapłaty: 276.55zł	Opłać rezerwację
Wycieczka 16 Kraj: Australia	Data wycieczki: 2021-06-27 Ilość zarezerwowanych miejsc: 2	Zapłacono: 1400.0 / 1400.0zł Pozostało do zapłaty: 0.0zł	Opłać rezerwację
Nowsza Wycieczka Kraj: Egipt	Data wycieczki: 2021-07-01 Ilość zarezerwowanych miejsc: 3	Zapłacono: 2100.0 / 2100.0zł Pozostało do zapłaty: 0.0zł	Opłać rezerwację
Wycieczka 5 Kraj: Rosja	Data wycieczki: 2021-07-01 Ilość zarezerwowanych miejsc: 1	Zapłacono: 200.0 / 200.0zł Pozostało do zapłaty: 0.0zł	Opłać rezerwację
Wycieczka 8 Kraj: Australia	Data wycieczki: 2021-07-15 Ilość zarezerwowanych miejsc: 2	Zapłacono: 400.0 / 900.0zł Pozostało do zapłaty: 500.0zł	Opłać rezerwację
Wycieczka 9 Kraj: Argentyna	Data wycieczki: 2021-07-15 Ilość zarezerwowanych miejsc: 2	Zapłacono: 230.0 / 800.0zł Pozostało do zapłaty: 570.0zł	Opłać rezerwację
Wycieczka 14 Kraj: Egipt	Data wycieczki: 2021-08-12 Ilość zarezerwowanych miejsc: 5	Zapłacono: 2250.0 / 2250.0zł Pozostało do zapłaty: 0.0zł	Opłać rezerwację

Dodawanie/Edycja wycieczek (FXMLAddTourController/FXMLEditTourController)



Nazwa wycieczki: Wycieczka 14

Kraj: Egipt

Data: 12.08.2021

Maksymalna ilosc miejsc: 15

Cena: 450.0

Opis: Lorem ipsum dolor sit amet, consectetur adipiscing elit. Pha

Zdjecie pogladowe

● Zdjecie z internetu ○ Zdjecie z komputera

https://i.pinimg.com/originals/70/a

7. Dokumentacja

Poniższa dokumentacja nie zawiera kodu widoków UI, pomocniczych singletonów, czy innych klas pomocniczych, niezwiązanych z funkcjonowaniem bazy danych.

7.1 Plik konfiguracyjny

Zawartość pliku hibernate.cfg.xml

```xml
<?xml version = "1.0" encoding = "utf-8"?>
<!DOCTYPE hibernate-configuration SYSTEM
       "http://www.hibernate.org/dtd/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
   <session-factory>
       <property name="hibernate.dialect">
           org.hibernate.dialect.Oracle8iDialect
       </property>
       <property name="hibernate.connection.driver_class">
           oracle.jdbc.OracleDriver</property>
       <property name="hibernate.connection.url">
           jdbc:oracle:thin:@dbmanage.lab.ki.agh.edu.pl:1521:DBMANAGE
       </property>
       <property name="hibernate.connection.username">
           BD_PALCZAK
       </property>
       <property name="hibernate.connection.password">
           GrzcbenelCnffjbeq123
       </property>
       <property name="hibernate.show_sql">true</property>
       <property name="hibernate.hbm2ddl.auto">
           update
       </property>



       <property name="hibernate.current_session_context_class">
           thread
       </property>

       <mapping class =
"agh.cs.projekt.models.ImageSource.ImageSource"/>
       <mapping class =
"agh.cs.projekt.models.ImageSource.LocalImageSource"/>
       <mapping class =
"agh.cs.projekt.models.ImageSource.HttpImageSource"/>
```

```
       <mapping class = "agh.cs.projekt.models.Tour"/>
       <mapping class = "agh.cs.projekt.models.Customer"/>
       <mapping class = "agh.cs.projekt.models.Rating"/>
       <mapping class = "agh.cs.projekt.models.Reservation"/>
       <mapping class = "agh.cs.projekt.models.Payment"/>
       <mapping class = "agh.cs.projekt.models.ApplicationUser"/>

   </session-factory>
</hibernate-configuration>
```

7.2 Modele bazodanowe w kodzie

a) Tour

```java
package agh.cs.projekt.models;

import agh.cs.projekt.services.DatabaseHolder;
import agh.cs.projekt.models.ImageSource.ImageSource;
import javafx.util.Pair;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

import javax.persistence.*;
import java.math.BigDecimal;
import java.sql.Date;
import java.util.List;

@Entity
public class Tour {

   @Id
   @GeneratedValue(strategy = GenerationType.AUTO)
   private int id;

   @Column(nullable = false)
   private String name;

   @Enumerated(EnumType.STRING)
   @Column(nullable = false)
   private CountryEnum country;

   @Column(nullable = false)
   private Date tourDate;

   private int maxPlaces;

   private float price;

   @Lob
   @Column(nullable = false)
   private String description;

   @ManyToOne(cascade = CascadeType.ALL)
   private ImageSource image;
```

```
    public Tour(){
        //required by Hibernate
    }

    public Tour(String name, CountryEnum country, Date date, int
maxPlaces, float price, String description, ImageSource image) {
        this.name = name;
        this.country = country;
        this.tourDate = date;
        this.maxPlaces = maxPlaces;
        this.price = price;
        this.description = description;
        this.image = image;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public CountryEnum getCountry() {
        return country;
    }

    public void setCountry(CountryEnum country) {
        this.country = country;
    }

    public Date getTourDate() {
        return tourDate;
    }

    public void setTourDate(Date tourDate) {
        this.tourDate = tourDate;
    }

    public int getMaxPlaces() {
        return maxPlaces;
    }

    public void setMaxPlaces(int maxPlaces) {
        this.maxPlaces = maxPlaces;
    }

    public float getPrice() {
        return price;
    }

    public void setPrice(float price) {
        this.price = price;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public ImageSource getImage() {
        return image;
    }

    public void setImage(ImageSource image) {
        this.image = image;
    }

    public int getAvailablePlaces(){ //returns a negative number on error
        try(Session session = DatabaseHolder.getInstance().getSession())
{
            Transaction transaction = session.beginTransaction();
            int result = getAvailablePlaces(session);
            transaction.commit();
            return result;
        } catch (Exception e){
            System.err.println("Error when fetching number of places");
            e.printStackTrace();
            return -1;
        }
    }

    public int getAvailablePlaces(Session session) { //returns a negative
number on error
```

```
        try {
            BigDecimal result = (BigDecimal)
session.createNativeQuery("select AVAILABLE_PLACES(:id) from
dual").setParameter("id", this.id).getSingleResult();
            System.out.println("Result " + result);
            return result.intValue();
        } catch (NoResultException e){
            return -1;
        }
    }

    //returns a pair of (average rating, total ratings), or null in case
of error
    public Pair<Double, Long> getRating(){
        try(Session session = DatabaseHolder.getInstance().getSession())
{
            Transaction transaction = session.beginTransaction();

            BigDecimal ratingsAmt = (BigDecimal)
session.createNativeQuery("select RATINGS_AMT(:id) from
dual").setParameter("id", this.id).getSingleResult();
            BigDecimal average = (BigDecimal)
session.createNativeQuery("select AVG_RATING(:id) from
dual").setParameter("id", this.id).getSingleResult();
            if (ratingsAmt == null) ratingsAmt = BigDecimal.valueOf(-1L);
            if (average == null) average = BigDecimal.valueOf(0.0);

            transaction.commit();
            return new Pair<>(average.doubleValue(),
ratingsAmt.longValue());
        } catch (Exception e){
            System.err.println("Error when fetching number of places");
            e.printStackTrace();
            return null;
        }
    }

    @Override
    public String toString() {
        return "Tour{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", country=" + country +
                ", tourDate=" + tourDate +
                ", maxPlaces=" + maxPlaces +
                ", price=" + price +
               ", description='" + description + '\'' +
               '}';
   }
}
```

b) ImageSource

```
package agh.cs.projekt.models.ImageSource;

import javafx.scene.image.Image;

import javax.persistence.*;
import java.io.IOException;

@Entity
@Inheritance(strategy = InheritanceType.JOINED)
public abstract class ImageSource {

   @Id
   @Column(nullable = false)
   protected String path;

   public ImageSource() {
       //required by hibernate
   }

   public abstract String getName() throws IOException;
   public abstract Image getImage() throws IOException;

}
```

c) HttpImageSource

```java
package agh.cs.projekt.models.ImageSource;

import javafx.embed.swing.SwingFXUtils;
import javafx.scene.image.Image;

import javax.imageio.ImageIO;
import javax.persistence.Entity;
import java.io.IOException;
import java.net.URL;

@Entity
public class HttpImageSource extends ImageSource {

   public HttpImageSource() {
       //required by hibernate
   }

   public HttpImageSource(String path) { //path is a "http:..." url
       this.path = path;
   }

   @Override
   public String getName() throws IOException {
       return new URL(path).toString();
   }

   @Override
   public Image getImage() throws IOException{
       return SwingFXUtils.toFXImage(ImageIO.read(new URL(path)), null);
   }

}
```

d) LocalImageSource

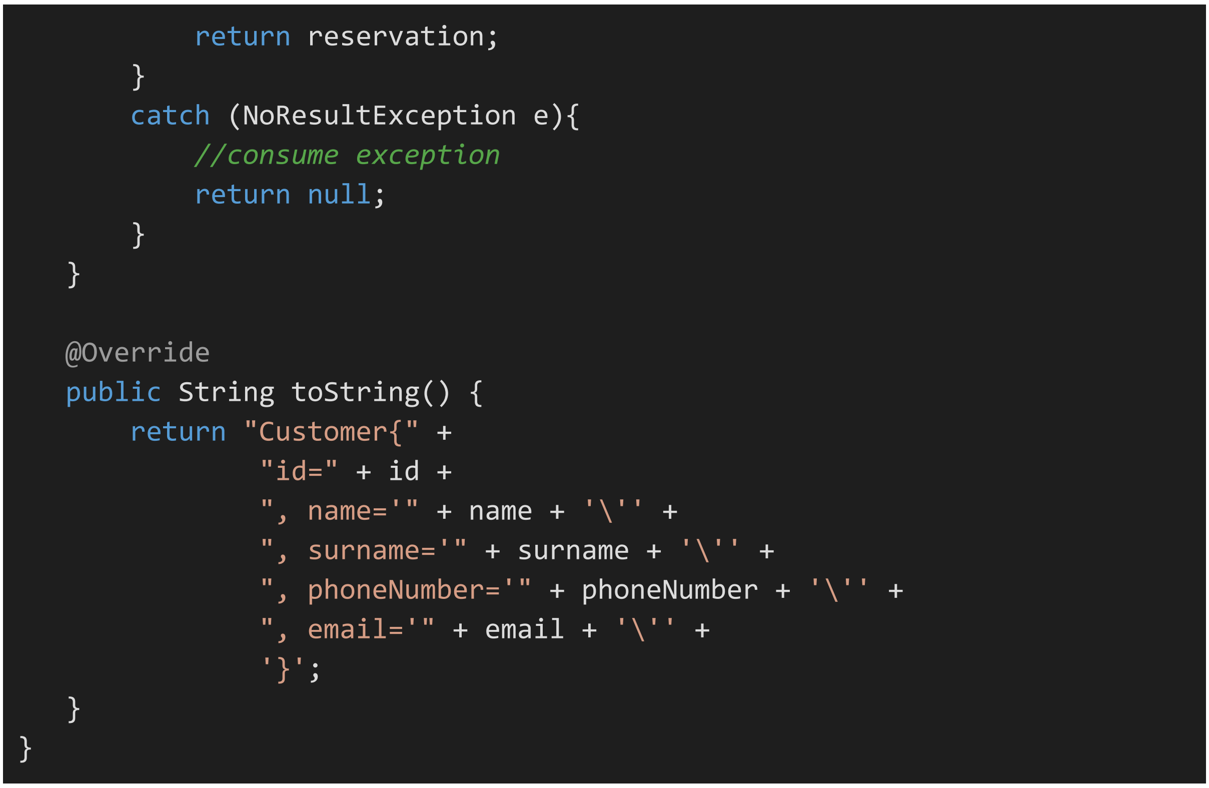
```java
package agh.cs.projekt.models.ImageSource;

import javafx.scene.image.Image;

import javax.imageio.ImageIO;
import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.Lob;
import java.awt.image.BufferedImage;
import java.io.ByteArrayInputStream;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.IOException;
import java.net.URISyntaxException;

@Entity
public class LocalImageSource extends ImageSource{

   @Lob
   @Column(nullable=false, columnDefinition="BLOB")
   protected byte[] image;

   public LocalImageSource() {
       //required by hibernate
   }

   public LocalImageSource(String path) { //path is an absolute path to the image (but now that the UI handles that, you should not worry about it)
       try {
           File file = new File(path);
           BufferedImage bImage = ImageIO.read(file);
           ByteArrayOutputStream bos = new ByteArrayOutputStream();
           ImageIO.write(bImage, "jpg", bos);
           this.image = bos.toByteArray();
           this.path = path;
       } catch (IOException e){
           System.err.println("Error when reading image");
           e.printStackTrace();
       }
   }

   @Override
   public String getName() throws IOException{
       return path;
```

```
    }

    @Override
    public Image getImage() throws IOException{
        return new Image(new ByteArrayInputStream(this.image));
    }

}
```

e) Customer

```
package agh.cs.projekt.models;

import agh.cs.projekt.services.DatabaseHolder;
import org.hibernate.Session;
import org.hibernate.Transaction;
import org.hibernate.query.Query;

import javax.persistence.*;
import java.io.Serializable;
import java.util.List;

@Entity
@Embeddable
public class Customer implements Serializable {

   @Id
   @GeneratedValue(strategy = GenerationType.AUTO)
   private int id;

   @Column(nullable = false)
   private String name;

   @Column(nullable = false)
   private String surname;

   @Column(nullable = false)
   private String phoneNumber;

   @Column(nullable = false)
   private String email;

   public Customer() {
       //required by Hibernate
   }

   public Customer(String name, String surname, String phoneNumber,
String email) {
       this.name = name;
       this.surname = surname;
       this.phoneNumber = phoneNumber;
       this.email = email;
   }

   public int getId() {
       return id;
```

```
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getSurname() {
        return surname;
    }

    public void setSurname(String surname) {
        this.surname = surname;
    }

    public String getPhoneNumber() {
        return phoneNumber;
    }

    public void setPhoneNumber(String phoneNumber) {
        this.phoneNumber = phoneNumber;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }

    public Reservation getReservationForTour(Tour tour){
        try(Session session = DatabaseHolder.getInstance().getSession())
{
            Transaction transaction = session.beginTransaction();

            Reservation reservation = session.createQuery("from
Reservation where tour = :tour and customer = :customer",
Reservation.class)
                    .setParameter("tour", tour)
                    .setParameter("customer", this).getSingleResult();

            transaction.commit();
```



```
            return reservation;
        }
        catch (NoResultException e){
            //consume exception
            return null;
        }
    }

    @Override
    public String toString() {
        return "Customer{" +
                "id=" + id +
                ", name='" + name + '\'' +
                ", surname='" + surname + '\'' +
                ", phoneNumber='" + phoneNumber + '\'' +
                ", email='" + email + '\'' +
                '}';
    }
}
```

f) ApplicationUser

```java
package agh.cs.projekt.models;

import javax.persistence.*;

@Entity
public class ApplicationUser {
   @Id
   @GeneratedValue(strategy = GenerationType.AUTO)
   private int id;

   @OneToOne
   @JoinColumn(nullable = false)
   private Customer customer;

   @Column(nullable = false)
   private String login;

   @Column(nullable = false)
   private String password;

   @Column(nullable = false)
   private String salt;

   @Column(nullable = false)
   private RoleEnum role;

   public ApplicationUser(){
       //required by Hibernate
   }

   public ApplicationUser(Customer customer, String login, String
password, String salt, RoleEnum role) {
       this.customer = customer;
       this.login = login;
       this.password = password;
       this.salt = salt;
       this.role = role;
   }

   public int getId() {
       return id;
   }

   public Customer getCustomer() {
       return customer;
```

```java
    }

    public void setCustomer(Customer customer) {
        this.customer = customer;
    }

    public String getLogin() {
        return login;
    }

    public void setLogin(String login) {
        this.login = login;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }

    public String getSalt() {
        return salt;
    }

    public void setSalt(String salt) {
        this.salt = salt;
    }

    public RoleEnum getRole() {
        return role;
    }

    public void setRole(RoleEnum role) {
        this.role = role;
    }

    @Override
    public String toString() {
        return "User{" +
                "id=" + id +
                ", customer=" + customer.getId() +
                ", login=" + login +
                ", password=" + password +
                ", salt=" + salt +
               ", role=" + role +
               "}";
   }

}
```

g) Reservation

(poniżej również kod ReservationID)

```
package agh.cs.projekt.models;

import agh.cs.projekt.services.DatabaseHolder;
import org.hibernate.Session;
import org.hibernate.Transaction;

import javax.persistence.*;
import java.math.BigDecimal;

@Entity
public class Reservation {

    @EmbeddedId
    private ReservationID reservationID;

    @MapsId("customerID")
    @ManyToOne(optional = false)
    @JoinColumns(value = {
            @JoinColumn(name = "customerID", referencedColumnName =
"id") })
    private Customer customer;

    @MapsId("tourID")
    @ManyToOne(optional = false)
    @JoinColumns(value = {
            @JoinColumn(name = "tourID", referencedColumnName = "id") })
    private Tour tour;

    private int reservedPlaces;

    public Reservation() {
        //required by Hibernate
        this.reservationID = new ReservationID();
    }

    public Reservation(Customer customer, Tour tour, int reservedPlaces)
{
        this.customer = customer;
        this.tour = tour;
        this.reservedPlaces = reservedPlaces;
        this.reservationID = new ReservationID();
    }

    public Customer getCustomer() {
        return customer;
    }

    public void setCustomer(Customer customer) {
        this.customer = customer;
    }

    public Tour getTour() {
        return tour;
    }

    public void setTour(Tour tour) {
        this.tour = tour;
    }

    public int getReservedPlaces() {
        return reservedPlaces;
    }

    public void setReservedPlaces(int reservedPlaces) {
        this.reservedPlaces = reservedPlaces;
    }

    // redundant, kept for compatibility
    public int getReservedAmount(){
        return getReservedPlaces();
    }

    public Reservation setPlacesAndPersist(int places){
        try (Session session =
DatabaseHolder.getInstance().getSession()){
            Transaction transaction = session.beginTransaction();
            setReservedPlaces(places);
            if (places > 0) {
                session.save(this);
                transaction.commit();
                return this;
            } else if (places == 0){
                session.delete(this);
                transaction.commit();
                return null;
            } else {
                throw new IllegalArgumentException("Places can't be
negative");
            }
        }
    }

    public Reservation changePlacesAndPersist(int placesDelta){
        return setPlacesAndPersist(this.getReservedPlaces() +
placesDelta);
    }

    //returns the rating object for this reservation, or null if no
reservation was made
    public Rating getRatingForReservation(){
        try (Session session =
DatabaseHolder.getInstance().getSession()){
            Transaction transaction = session.beginTransaction();

            Rating rating = session.createQuery("from Rating where
reservation = :reservation", Rating.class)
                    .setParameter("reservation", this)
                    .getSingleResult();
            transaction.commit();

            return rating;
        } catch (NoResultException e){
            //consume error
            return null;
        }
    }

    //adds a rating to a reservation, returns the new rating object, or
null if value was 0 (rating was deleted)
    public Rating setRating(int value) throws Exception {
        Rating currentRating = this.getRatingForReservation();

        try(Session session =
DatabaseHolder.getInstance().getSession()){
            Transaction transaction = session.beginTransaction();

            if (currentRating == null){
                //user hasn't rated yet
                if (value == 0){
                    //user doesn't want to add a rating
                    transaction.commit();
                    return null;
                } else {
                    //add new rating
                    Rating rating = new Rating(this, value);
                    session.save(rating);
                    transaction.commit();
                    return rating;
                }
            } else {
                //user wants to change the rating
                if (value == 0) {
                    //user wants to remove the rating
                    session.delete(currentRating);
                    transaction.commit();
                    return null;
                } else {
                    //user wants to alter the rating
                    currentRating.setRating(value);
                    session.update(currentRating);
                    transaction.commit();
                    return currentRating;
                }
            }
        }
    }

    public float getCurrentlyPaidAmount() {
        Session session = DatabaseHolder.getInstance().getSession();
        session.beginTransaction();

        System.out.println("tour_id: " +
this.reservationID.getTourID());
        System.out.println("customer_id: " +
this.reservationID.getCustomerID());
        BigDecimal currentlyPaidAmount = (BigDecimal)
session.createNativeQuery(
                "select CURRENTLY_PAID(:tour_id, :customer_id) from
dual"
        ).setParameter("tour_id", this.reservationID.getTourID())
                .setParameter("customer_id",
this.reservationID.getCustomerID())
                .getSingleResult();
        session.getTransaction().commit();
        return currentlyPaidAmount.floatValue();
    }

    public float getTotalPrice() {
        return this.getReservedPlaces() * this.getTour().getPrice();
    }

    public float getRemainingPayment() {
        return this.getTotalPrice() - this.getCurrentlyPaidAmount();
    }

    @Override
    public String toString() {
        return "Reservation{" +
                ", customer=" + customer +
                ", tour=" + tour +
                ", reservedPlaces=" + reservedPlaces +
                '}';
    }
}
```

```
package agh.cs.projekt.models;

import javax.persistence.Embeddable;
import java.io.Serializable;

@Embeddable
public class ReservationID implements Serializable {
    private Integer customerID;
    private Integer tourID;

    public ReservationID() {
        //required by Hibernate
    }

    public Integer getCustomerID() {
        return customerID;
    }

    public void setCustomerID(Integer customerID) {
        this.customerID = customerID;
    }

    public Integer getTourID() {
        return tourID;
    }

    public void setTourID(Integer tourID) {
        this.tourID = tourID;
    }
}
```

h) Payment

```
package agh.cs.projekt.models;

import javax.persistence.*;
import java.sql.Date;

@Entity
public class Payment {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private int id;

    @ManyToOne
    private Reservation reservation;

    @Column(nullable = false)
    private Date paymentDate;

    private float amount;

    public Payment() {
        //required by Hibernate
    }

    public Payment(Reservation reservation, Date paymentDate, float
amount) {
        this.reservation = reservation;
        this.paymentDate = paymentDate;
        this.amount = amount;
    }

    public int getId() {
        return id;
    }

    public Reservation getReservation() {
        return reservation;
    }

    public void setReservation(Reservation reservation) {
        this.reservation = reservation;
    }

    public Date getPaymentDate() {
        return paymentDate;
    }

    public void setPaymentDate(Date paymentDate) {
        this.paymentDate = paymentDate;
    }

    public float getAmount() {
        return amount;
    }

    public void setAmount(float amount) {
        this.amount = amount;
    }

    @Override
    public String toString() {
        return "Payment{" +
                "id=" + id +
                ", reservation=" + reservation +
                ", paymentDate=" + paymentDate +
                ", amount=" + amount +
                '}';
    }

}
```

i) Rating

```java
package agh.cs.projekt.models;

import org.hibernate.annotations.OnDelete;
import org.hibernate.annotations.OnDeleteAction;

import javax.persistence.Entity;
import javax.persistence.Id;
import javax.persistence.OneToOne;
import java.io.Serializable;

@Entity
public class Rating implements Serializable {

    @Id
    @OneToOne()
    @OnDelete(action = OnDeleteAction.CASCADE)
    private Reservation reservation;

    private int rating;

    public Rating() {
        //required by Hibernate
    }

    public Rating(Reservation reservation, int rating) {
        this.reservation = reservation;
        this.rating = rating;
    }

    public Reservation getReservation() {
        return reservation;
    }

    public void setReservation(Reservation reservation) {
        this.reservation = reservation;
    }

    public int getRating() {
        return rating;
    }

    public void setRating(int rating) {
        this.rating = rating;
    }
    @Override
    public String toString() {
        return "Rating{" +
                "reservation=" + reservation +
                ", rating=" + rating +
                '}';
    }
}
```

7.3 Funkcje zdefiniowane w bazie danych

available_places(queried_id) - zwraca ilość wolnych miejsc dla wycieczki o ID queried_id

```
create FUNCTION available_places(
   queried_id NUMBER
)
   RETURN NUMBER
   IS
   available NUMBER := 0;
   total NUMBER := 0;
   taken NUMBER := 0;
BEGIN
   select MAXPLACES into total from TOUR where ID = queried_id;
   select SUM(RESERVEDPLACES) into taken from RESERVATION where TOURID =
queried_id;
   available := total -  COALESCE(taken, 0);

   RETURN available;
END;
/
```

avg_rating(queried_id) - zwraca średnią ocenę wycieczki o ID queried_id

```
create FUNCTION avg_rating(
   queried_id NUMBER
)
   RETURN NUMBER
   IS
   average NUMBER := 0;
BEGIN
   select AVG(R.RATING)
   into average
   from RATING R
   inner join TOUR T on R.RESERVATION_TOURID = T.ID
   where T.ID = queried_id;

   RETURN average;
END;
/
```

ratings_amt(queried_id) - zwraca ilość ocen dla wycieczki o ID queried_id

```
create FUNCTION ratings_amt(
   queried_id NUMBER
)
   RETURN NUMBER
   IS
   total NUMBER := 0;
BEGIN
   select COUNT(*)
   into total
   from RATING
   inner join TOUR T on RATING.RESERVATION_TOURID = T.ID
   where T.ID = queried_id;

   RETURN total;
END;
/
```

7.4 Kod wypełniający bazę danych

Uwaga! Nie udało nam się zrzucić wartości przechowywanych jako blob do poleceń INSERT INTO, stąd poniżej nie przedstawiamy kodu generującego tabeli LocalImageSource.

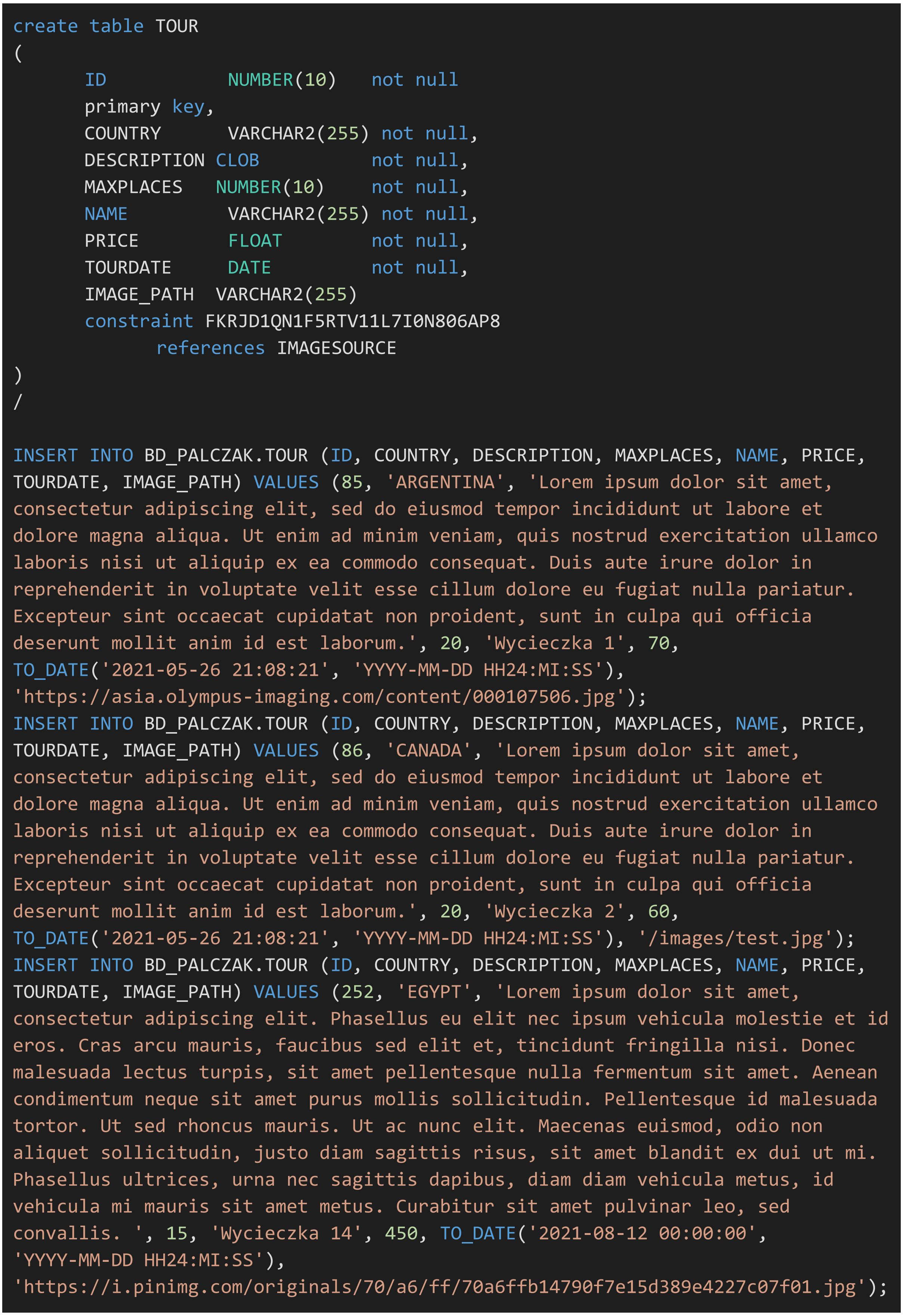
Tabela Customer

```
create table CUSTOMER
(
       ID           NUMBER(10)   not null
       primary key,
       EMAIL        VARCHAR2(255) not null,
       NAME         VARCHAR2(255) not null,
       PHONENUMBER VARCHAR2(255) not null,
       SURNAME      VARCHAR2(255) not null
)
/

INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(257, 'anowak@mail.com', 'Adam', '234567891', 'Nowak');
INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(261, 'jkwiatkowska@mail.com', 'Julia', '456789123', 'Kwiatkowska');
INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(265, 'wwojciechowska@mail.com', 'Wiktoria', '678912345', 'Wojciechowska');
INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(83, 'dewkong@example.com', 'Krystian', '123123123', 'Adminowy');
INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(61, 'wojtek@example.com', 'Wojtek', '123456789', 'Piechaczek');
INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(101, 'marcin@example.com', 'Marcin', '111222333', 'Palczak');
INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(255, 'jkowalski@mail.com', 'Jan', '123456789', 'Kowalski');
INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(259, 'jwisniewski@mail.com', 'Jakub', '345678912', 'Kowalczyk');
INSERT INTO BD_PALCZAK.CUSTOMER (ID, EMAIL, NAME, PHONENUMBER, SURNAME) VALUES
(263, 'apiotrowska@mail.com', 'Aleksandra', '567891234', 'Piotrowska');
```

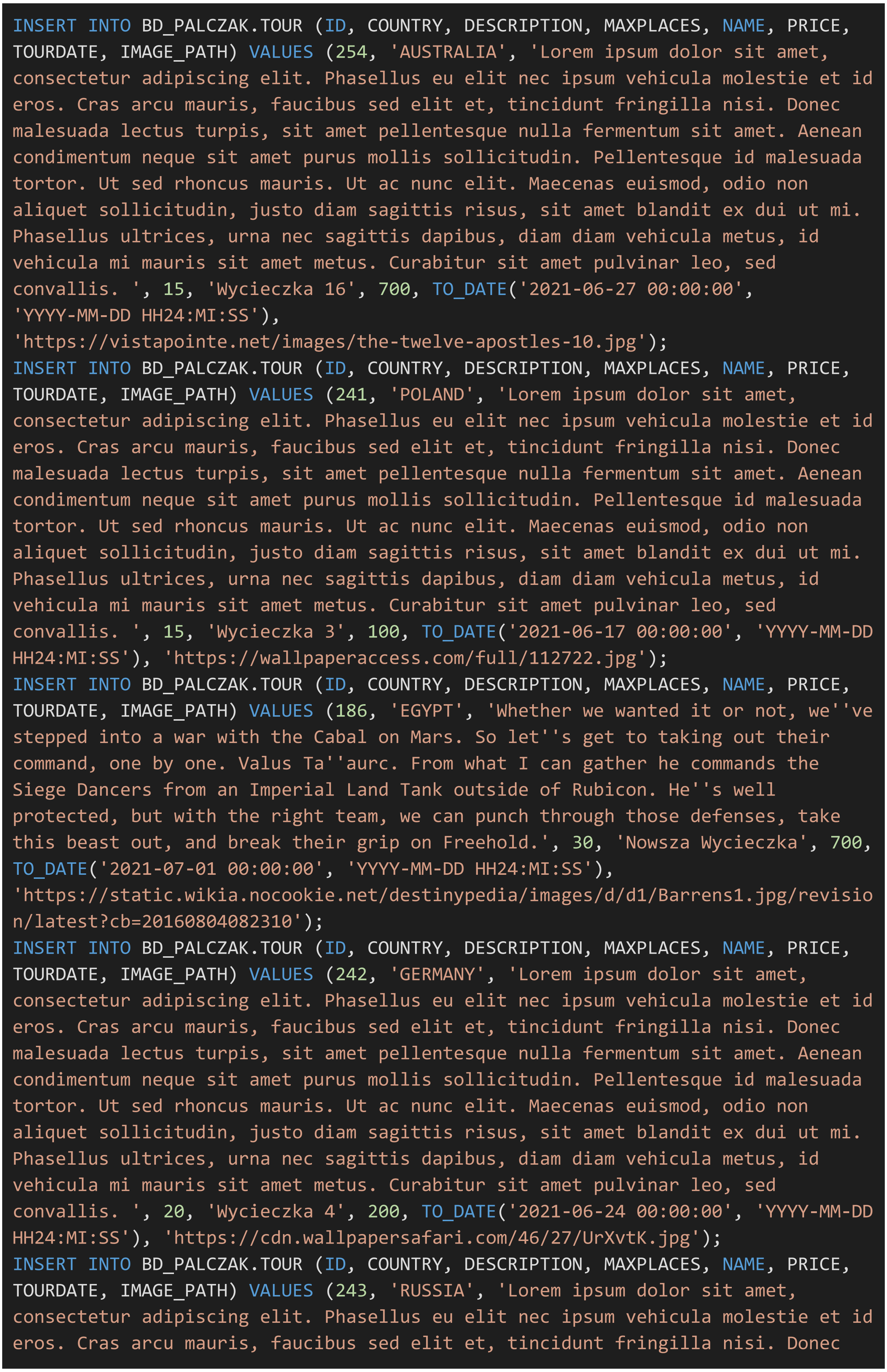
Tabela ApplicationUser

```
create table APPLICATIONUSER
(
       ID           NUMBER(10)   not null
       primary key,
       LOGIN        VARCHAR2(255) not null,
       PASSWORD     VARCHAR2(255) not null,
       ROLE         NUMBER(10)   not null,
       SALT         VARCHAR2(255) not null,
       CUSTOMER_ID NUMBER(10)    not null
       constraint FK8W7337ANMMMES1TIH2RI4F8WH
             references CUSTOMER
)
/

INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (258, 'Login2',
'JX0oJmsvnurBSLrs4F79E8pLTcXdTWo/tw5oQBev2+4=', 0, 'EYZo4ztbQylw7dOVOMff', 257);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (262, 'Login4',
'VwXosv89OG6zLJZv6G9m9j85KjUWpk+OZ4XdmlMfDcA=', 0, 'AnMb1YYfOHZbMvo6Uok2', 261);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (266, 'Login6',
'nzO282x7ig7Vg0Y9CfHS4pVXipTsjyOkFvZnxorlXBM=', 0, 'wzGOxjwCxAkmar7pijNT', 265);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (84, 'Dewkong',
'g9UnIw2+est0mkGg+Qpk4FTGvdedpvITfVZ90obEvRo=', 1, 'cOMbp6JOwgSSyQg9ra1b', 83);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (62, 'Wojtek',
'UDe8TjcGZTtbir1r7wvqsfZX+jo1B+j/I/3k8X1Hd6s=', 1, 'xOxU4XTEq61g2vcelfaf', 61);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (102, 'Garion',
'pjDqmB/Mz/J9hjbaBYL4nj/3t1lO9dnAX2BIBNjVeBY=', 1, '7xrouVtTUH4ZzHgZPWke', 101);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (256, 'Login1',
'DEpng4qJRs8LGVN7c4gViui8/3Sw03Qs1GZtqTJKXgI=', 0, '7wiI5pMqJGi4f8evfRdj', 255);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (260, 'Login3',
'yRBV0J98bfXTHCZ1vBvigMyttZsuN9K5VmZkPhUhuGc=', 0, 'uCi7yXGy70gTIfzMtdYL', 259);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (264, 'Login5',
'+RcBZHqDxpeTQwP7k7xO/tcDZjbbegI0DngloRwjNUU=', 0, 'v4yj1uUEM3U8q1QDdKSm', 263);
INSERT INTO BD_PALCZAK.APPLICATIONUSER (ID, LOGIN, PASSWORD, ROLE, SALT,
CUSTOMER_ID) VALUES (362, 'admin',
'1dF8PzJ1nGicBZCCmlplvtTdBy8YHEYSXtzIaVZPG18=', 1, 'JLadFRxFn4YqFQq5b0KS', 361);
```

Tabela Tour



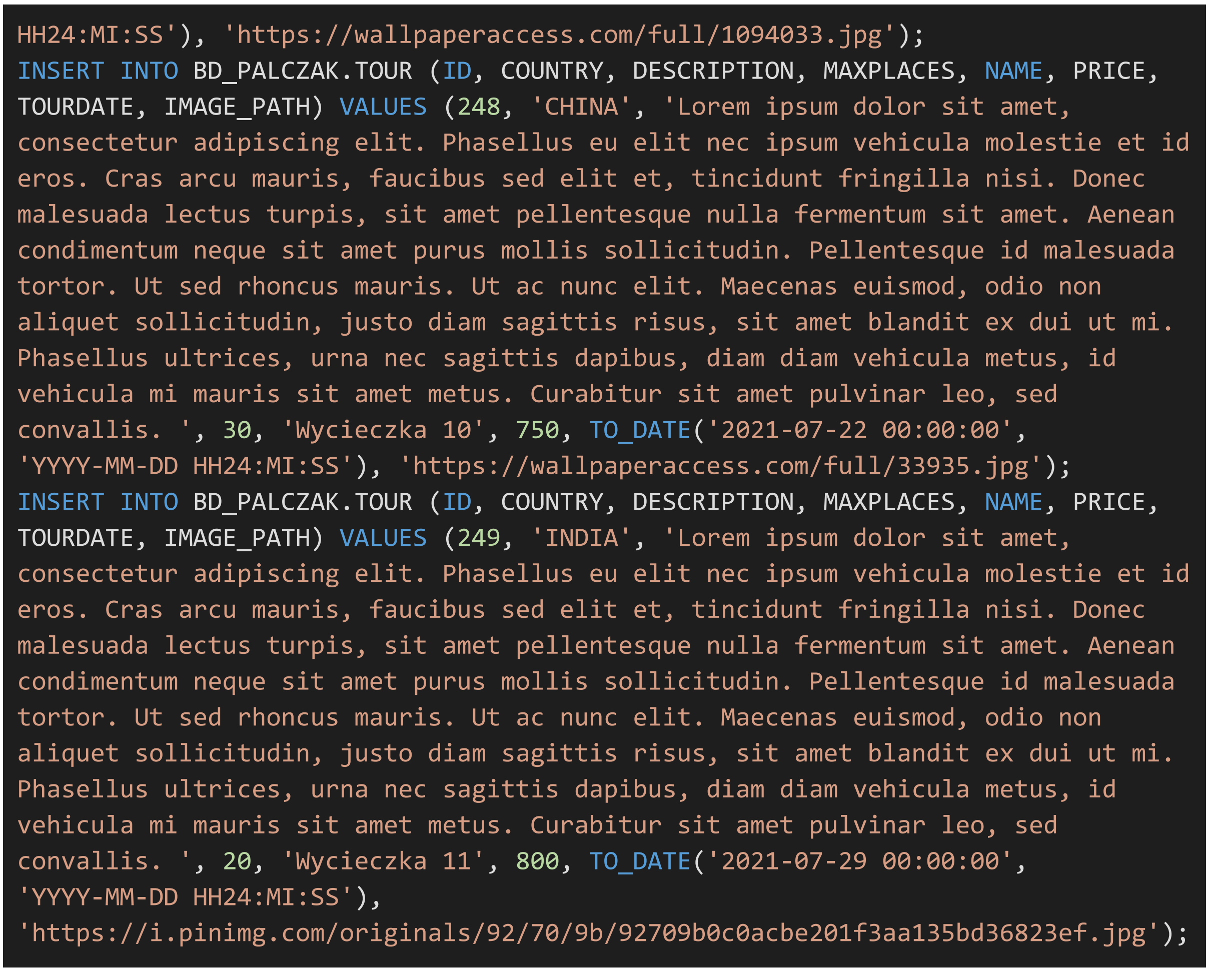
```
create table TOUR
(
       ID            NUMBER(10)   not null
       primary key,
       COUNTRY       VARCHAR2(255) not null,
       DESCRIPTION CLOB          not null,
       MAXPLACES   NUMBER(10)    not null,
       NAME          VARCHAR2(255) not null,
       PRICE         FLOAT         not null,
       TOURDATE      DATE          not null,
       IMAGE_PATH  VARCHAR2(255)
       constraint FKRJD1QN1F5RTV11L7I0N806AP8
             references IMAGESOURCE
)
/

INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (85, 'ARGENTINA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.', 20, 'Wycieczka 1', 70,
TO_DATE('2021-05-26 21:08:21', 'YYYY-MM-DD HH24:MI:SS'),
'https://asia.olympus-imaging.com/content/000107506.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (86, 'CANADA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.', 20, 'Wycieczka 2', 60,
TO_DATE('2021-05-26 21:08:21', 'YYYY-MM-DD HH24:MI:SS'), '/images/test.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (252, 'EGYPT', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 15, 'Wycieczka 14', 450, TO_DATE('2021-08-12 00:00:00',
'YYYY-MM-DD HH24:MI:SS'),
'https://i.pinimg.com/originals/70/a6/ff/70a6ffb14790f7e15d389e4227c07f01.jpg');
```



```
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (254, 'AUSTRALIA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 15, 'Wycieczka 16', 700, TO_DATE('2021-06-27 00:00:00',
'YYYY-MM-DD HH24:MI:SS'),
'https://vistapointe.net/images/the-twelve-apostles-10.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (241, 'POLAND', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 15, 'Wycieczka 3', 100, TO_DATE('2021-06-17 00:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'https://wallpaperaccess.com/full/112722.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (186, 'EGYPT', 'Whether we wanted it or not, we''ve
stepped into a war with the Cabal on Mars. So let''s get to taking out their
command, one by one. Valus Ta''aurc. From what I can gather he commands the
Siege Dancers from an Imperial Land Tank outside of Rubicon. He''s well
protected, but with the right team, we can punch through those defenses, take
this beast out, and break their grip on Freehold.', 30, 'Nowsza Wycieczka', 700,
TO_DATE('2021-07-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS'),
'https://static.wikia.nocookie.net/destinypedia/images/d/d1/Barrens1.jpg/revisio
n/latest?cb=20160804082310');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (242, 'GERMANY', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 20, 'Wycieczka 4', 200, TO_DATE('2021-06-24 00:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'https://cdn.wallpapersafari.com/46/27/UrXvtK.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (243, 'RUSSIA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
```

```
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 15, 'Wycieczka 5', 200, TO_DATE('2021-07-01 00:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'https://wallpaperaccess.com/full/401341.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (246, 'AUSTRALIA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 35, 'Wycieczka 8', 450, TO_DATE('2021-07-15 00:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'https://wallpaperaccess.com/full/108054.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (161, 'USA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et
dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco
laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in
reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia
deserunt mollit anim id est laborum.', 32, 'Słoneczne Hawaje', 210,
TO_DATE('2021-08-13 00:00:00', 'YYYY-MM-DD HH24:MI:SS'),
'https://empire-s3-production.bobvila.com/articles/wp-content/uploads/2020/06/Ty
pes-of-Palm-Trees.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (247, 'ARGENTINA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 20, 'Wycieczka 9', 400, TO_DATE('2021-07-15 00:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'https://wallpaperaccess.com/full/196893.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (251, 'UKRAINE', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
```

```
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 20, 'Wycieczka 13', 150, TO_DATE('2021-08-12 00:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'https://wallpaperaccess.com/full/1330367.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (250, 'SLOVAKIA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 25, 'Wycieczka 12', 250, TO_DATE('2021-08-05 00:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'https://wallpaperaccess.com/full/321742.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (253, 'USA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 15, 'Wycieczka 15', 550, TO_DATE('2021-08-19 00:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'https://wallpaperaccess.com/full/329352.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (244, 'CANADA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 10, 'Wycieczka 6', 250, TO_DATE('2021-07-08 00:00:00', 'YYYY-MM-DD
HH24:MI:SS'), 'https://wallpaperaccess.com/full/23618.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (245, 'SWITZERLAND', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 25, 'Wycieczka 7', 350, TO_DATE('2021-06-10 00:00:00', 'YYYY-MM-DD
```



```
HH24:MI:SS'), 'https://wallpaperaccess.com/full/1094033.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (248, 'CHINA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 30, 'Wycieczka 10', 750, TO_DATE('2021-07-22 00:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'https://wallpaperaccess.com/full/33935.jpg');
INSERT INTO BD_PALCZAK.TOUR (ID, COUNTRY, DESCRIPTION, MAXPLACES, NAME, PRICE,
TOURDATE, IMAGE_PATH) VALUES (249, 'INDIA', 'Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Phasellus eu elit nec ipsum vehicula molestie et id
eros. Cras arcu mauris, faucibus sed elit et, tincidunt fringilla nisi. Donec
malesuada lectus turpis, sit amet pellentesque nulla fermentum sit amet. Aenean
condimentum neque sit amet purus mollis sollicitudin. Pellentesque id malesuada
tortor. Ut sed rhoncus mauris. Ut ac nunc elit. Maecenas euismod, odio non
aliquet sollicitudin, justo diam sagittis risus, sit amet blandit ex dui ut mi.
Phasellus ultrices, urna nec sagittis dapibus, diam diam vehicula metus, id
vehicula mi mauris sit amet metus. Curabitur sit amet pulvinar leo, sed
convallis. ', 20, 'Wycieczka 11', 800, TO_DATE('2021-07-29 00:00:00',
'YYYY-MM-DD HH24:MI:SS'),
'https://i.pinimg.com/originals/92/70/9b/92709b0c0acbe201f3aa135bd36823ef.jpg');
```

Tabela ImageSource

```
create table IMAGESOURCE
(
	PATH VARCHAR2(255) not null
	primary key
)
/

INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('/images/test.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('C:\Users\User\Downloads\pexels-pixabay-210243.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('C:\Users\User\Downloads\verynicelandscape.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://asia.olympus-imaging.com/content/000107506.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://cdn.wallpapersafari.com/46/27/UrXvtK.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://empire-s3-production.bobvila.com/articles/wp-content/uploads/2020/06/Types-of-Palm-Trees.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://i.pinimg.com/originals/70/a6/ff/70a6ffb14790f7e15d389e4227c07f01.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://i.pinimg.com/originals/92/70/9b/92709b0c0acbe201f3aa135bd36823ef.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://i.pinimg.com/originals/98/01/84/980184fe0897895825bb2e751d8007f7.gif');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://isorepublic.com/wp-content/uploads/2018/11/egypt-1100x733.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://picsum.photos/1600/900');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://picsum.photos/800/450');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://static.wikia.nocookie.net/destinypedia/images/7/71/ValleyoftheKings.jpg/revision/latest?cb=20160804083216');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://static.wikia.nocookie.net/destinypedia/images/d/d1/Barrens1.jpg/revision/latest?cb=20160804082310');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://vistapointe.net/images/the-twelve-apostles-10.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://wallpaperaccess.com/full/108054.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://wallpaperaccess.com/full/1094033.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES ('https://wallpaperaccess.com/full/112722.jpg');
```

```sql
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/1330367.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/196893.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/23618.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/321742.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/329352.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/33935.jpg');
INSERT INTO BD_PALCZAK.IMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/401341.jpg');
```

Tabela HttpImageSource

```
create table HTTPIMAGESOURCE
(
        PATH VARCHAR2(255) not null
        primary key
        constraint FKP0HQBW9LB3NQ7BHR7GRO976SJ
                references IMAGESOURCE
)
/

INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://asia.olympus-imaging.com/content/000107506.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://cdn.wallpapersafari.com/46/27/UrXvtK.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://empire-s3-production.bobvila.com/articles/wp-content/uploads/2020/06/T
ypes-of-Palm-Trees.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://i.pinimg.com/originals/70/a6/ff/70a6ffb14790f7e15d389e4227c07f01.jpg')
;
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://i.pinimg.com/originals/92/70/9b/92709b0c0acbe201f3aa135bd36823ef.jpg')
;
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://i.pinimg.com/originals/98/01/84/980184fe0897895825bb2e751d8007f7.gif')
;
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://isorepublic.com/wp-content/uploads/2018/11/egypt-1100x733.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://picsum.photos/1600/900');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://picsum.photos/800/450');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://static.wikia.nocookie.net/destinypedia/images/7/71/ValleyoftheKings.jp
g/revision/latest?cb=20160804083216');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://static.wikia.nocookie.net/destinypedia/images/d/d1/Barrens1.jpg/revisi
on/latest?cb=20160804082310');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://vistapointe.net/images/the-twelve-apostles-10.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/108054.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/1094033.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/112722.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/1330367.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
```

```
('https://wallpaperaccess.com/full/196893.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/23618.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/321742.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/329352.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/33935.jpg');
INSERT INTO BD_PALCZAK.HTTPIMAGESOURCE (PATH) VALUES
('https://wallpaperaccess.com/full/401341.jpg');
```

Tabela Reservation

```
create table RESERVATION
(
	CUSTOMERID   NUMBER(10) not null
	constraint FKSKP8DB8KIVB03AKRY149SKSDR
		references CUSTOMER,
	TOURID       NUMBER(10) not null
	constraint FKDIS794C4BKE6QUUH8BI3WPPDX
		references TOUR,
	RESERVEDPLACES NUMBER(10) not null,
	primary key (CUSTOMERID, TOURID)
)
/

INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(101, 161, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(61, 161, 4);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(101, 186, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(101, 86, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 252, 5);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 241, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 186, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 246, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 161, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 249, 4);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 186, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 161, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 245, 5);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 244, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(259, 186, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(259, 241, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(259, 253, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
```

```
(261, 248, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(261, 252, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 246, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 253, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 245, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(265, 252, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 243, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(265, 186, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 242, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(83, 161, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 254, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 243, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(255, 247, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 252, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 242, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 248, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(257, 246, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(259, 242, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(259, 249, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(259, 251, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(261, 186, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(261, 254, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(261, 242, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(261, 161, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(261, 249, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 186, 3);
```

```sql
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 242, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 244, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 252, 1);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(263, 254, 2);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(265, 243, 3);
INSERT INTO BD_PALCZAK.RESERVATION (CUSTOMERID, TOURID, RESERVEDPLACES) VALUES
(265, 244, 3);
```

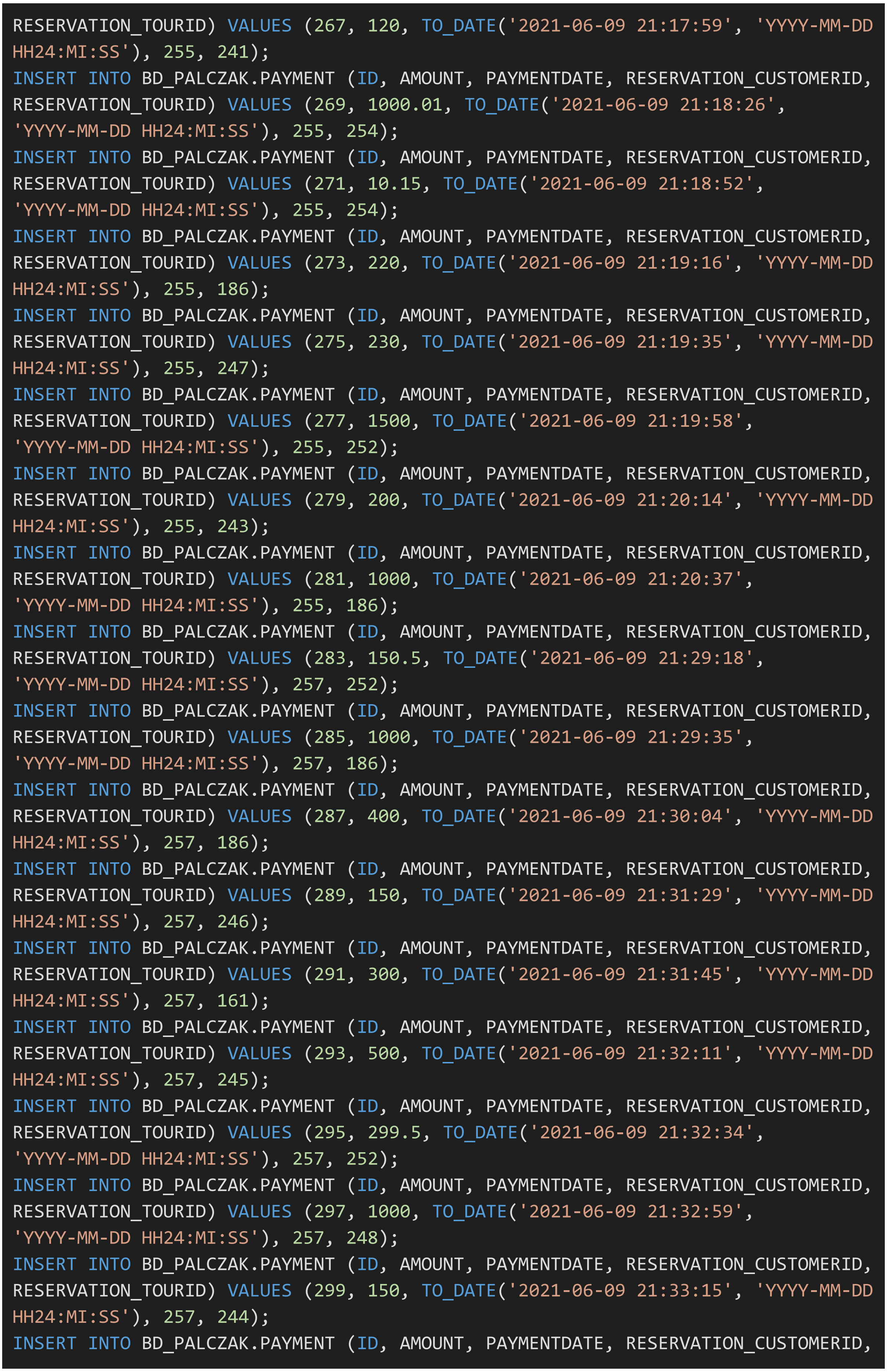
Tabela Rating

```
create table RATING
(
	RATING              NUMBER(10) not null,
	RESERVATION_CUSTOMERID NUMBER(10) not null,
	RESERVATION_TOURID  NUMBER(10) not null,
	primary key (RESERVATION_CUSTOMERID, RESERVATION_TOURID),
	constraint FKPBN5QRJ5SRSP31WJ7RRW2C2LL
	foreign key (RESERVATION_CUSTOMERID, RESERVATION_TOURID) references
RESERVATION
             on delete cascade
)
/

INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 255, 252);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 61, 161);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 101, 186);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (3, 255, 241);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 255, 186);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (2, 255, 246);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 255, 161);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 257, 249);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 257, 186);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 257, 161);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 257, 245);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 257, 244);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 259, 186);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 259, 241);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 259, 253);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 261, 248);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 261, 252);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 263, 246);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 263, 253);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (3, 263, 245);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 263, 243);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 265, 186);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 265, 252);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (3, 255, 254);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 255, 242);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 255, 243);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (3, 255, 247);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (2, 257, 252);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (1, 257, 242);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 257, 248);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 257, 246);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (3, 259, 242);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 259, 249);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 261, 186);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 261, 254);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (1, 261, 242);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 261, 161);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 261, 249);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 263, 186);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 263, 242);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 263, 252);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (5, 263, 254);
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 265, 243);
```

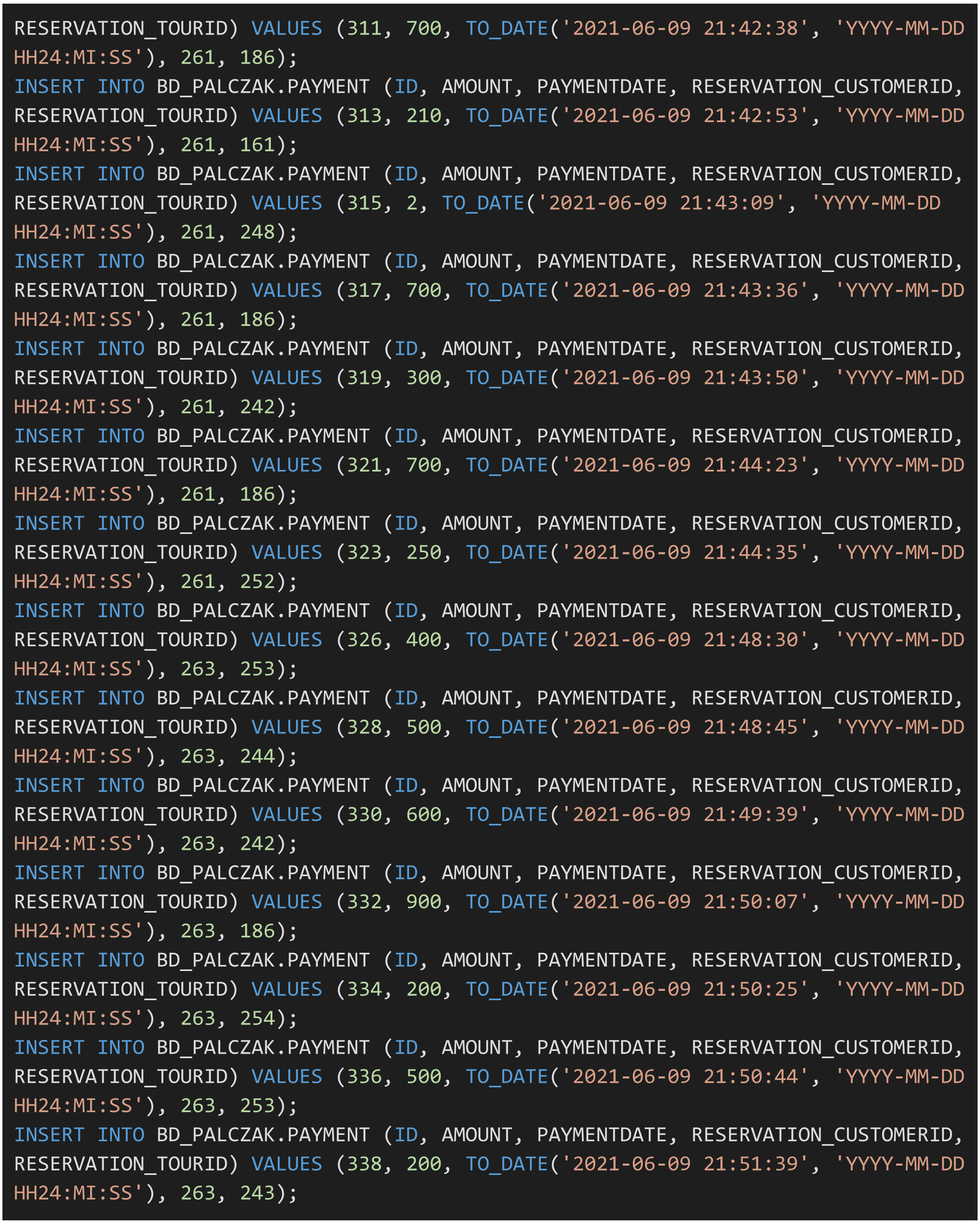
```
INSERT INTO BD_PALCZAK.RATING (RATING, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (4, 265, 244);
```

Tabela Payment

```
create table PAYMENT
(
       ID                  NUMBER(10) not null
       primary key,
       AMOUNT              FLOAT        not null,
       PAYMENTDATE         DATE         not null,
       RESERVATION_CUSTOMERID NUMBER(10),
       RESERVATION_TOURID  NUMBER(10),
       constraint FKJO7857VS5QBSN1E4HTWMA96NE
       foreign key (RESERVATION_CUSTOMERID, RESERVATION_TOURID) references
RESERVATION
)
/

INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (229, 0.11, TO_DATE('2021-06-08 13:04:50',
'YYYY-MM-DD HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (230, 1825.09, TO_DATE('2021-06-08 13:06:10',
'YYYY-MM-DD HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (231, 210, TO_DATE('2021-06-08 13:56:40', 'YYYY-MM-DD
HH24:MI:SS'), 101, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (232, 100, TO_DATE('2021-06-08 13:59:47', 'YYYY-MM-DD
HH24:MI:SS'), 101, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (233, 110, TO_DATE('2021-06-08 13:59:57', 'YYYY-MM-DD
HH24:MI:SS'), 101, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (268, 123.45, TO_DATE('2021-06-09 21:18:12',
'YYYY-MM-DD HH24:MI:SS'), 255, 242);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (270, 150.45, TO_DATE('2021-06-09 21:18:37',
'YYYY-MM-DD HH24:MI:SS'), 255, 254);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (272, 239.39, TO_DATE('2021-06-09 21:19:03',
'YYYY-MM-DD HH24:MI:SS'), 255, 254);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (274, 400, TO_DATE('2021-06-09 21:19:26', 'YYYY-MM-DD
HH24:MI:SS'), 255, 246);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (276, 630, TO_DATE('2021-06-09 21:19:50', 'YYYY-MM-DD
HH24:MI:SS'), 255, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (278, 750, TO_DATE('2021-06-09 21:20:04', 'YYYY-MM-DD
HH24:MI:SS'), 255, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (280, 880, TO_DATE('2021-06-09 21:20:30', 'YYYY-MM-DD
HH24:MI:SS'), 255, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (282, 200, TO_DATE('2021-06-09 21:29:03', 'YYYY-MM-DD
HH24:MI:SS'), 257, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (284, 400, TO_DATE('2021-06-09 21:29:25', 'YYYY-MM-DD
HH24:MI:SS'), 257, 242);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (286, 500, TO_DATE('2021-06-09 21:29:56', 'YYYY-MM-DD
HH24:MI:SS'), 257, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (288, 200, TO_DATE('2021-06-09 21:30:10', 'YYYY-MM-DD
HH24:MI:SS'), 257, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (290, 2000, TO_DATE('2021-06-09 21:31:36',
'YYYY-MM-DD HH24:MI:SS'), 257, 249);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (292, 100, TO_DATE('2021-06-09 21:31:53', 'YYYY-MM-DD
HH24:MI:SS'), 257, 244);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (294, 130, TO_DATE('2021-06-09 21:32:25', 'YYYY-MM-DD
HH24:MI:SS'), 257, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (296, 1000, TO_DATE('2021-06-09 21:32:51',
'YYYY-MM-DD HH24:MI:SS'), 257, 248);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (298, 250, TO_DATE('2021-06-09 21:33:05', 'YYYY-MM-DD
HH24:MI:SS'), 257, 248);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (300, 300, TO_DATE('2021-06-09 21:33:20', 'YYYY-MM-DD
HH24:MI:SS'), 257, 246);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (301, 777, TO_DATE('2021-06-09 21:33:58', 'YYYY-MM-DD
HH24:MI:SS'), 259, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (302, 777, TO_DATE('2021-06-09 21:37:22', 'YYYY-MM-DD
HH24:MI:SS'), 259, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (303, 300, TO_DATE('2021-06-09 21:37:28', 'YYYY-MM-DD
HH24:MI:SS'), 259, 251);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (304, 100, TO_DATE('2021-06-09 21:37:35', 'YYYY-MM-DD
HH24:MI:SS'), 259, 241);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (305, 100, TO_DATE('2021-06-09 21:37:42', 'YYYY-MM-DD
HH24:MI:SS'), 259, 241);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (306, 546, TO_DATE('2021-06-09 21:37:49', 'YYYY-MM-DD
HH24:MI:SS'), 259, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (307, 500, TO_DATE('2021-06-09 21:37:57', 'YYYY-MM-DD
HH24:MI:SS'), 259, 249);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (308, 300, TO_DATE('2021-06-09 21:38:14', 'YYYY-MM-DD
HH24:MI:SS'), 259, 249);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (309, 200, TO_DATE('2021-06-09 21:38:20', 'YYYY-MM-DD
HH24:MI:SS'), 259, 242);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (310, 400, TO_DATE('2021-06-09 21:38:27', 'YYYY-MM-DD
HH24:MI:SS'), 259, 253);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (312, 200, TO_DATE('2021-06-09 21:42:46', 'YYYY-MM-DD
HH24:MI:SS'), 261, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (314, 200, TO_DATE('2021-06-09 21:43:02', 'YYYY-MM-DD
HH24:MI:SS'), 261, 249);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (316, 248, TO_DATE('2021-06-09 21:43:17', 'YYYY-MM-DD
HH24:MI:SS'), 261, 248);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (318, 500, TO_DATE('2021-06-09 21:43:44', 'YYYY-MM-DD
HH24:MI:SS'), 261, 254);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (320, 210, TO_DATE('2021-06-09 21:44:14', 'YYYY-MM-DD
HH24:MI:SS'), 261, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (322, 100, TO_DATE('2021-06-09 21:44:29', 'YYYY-MM-DD
HH24:MI:SS'), 261, 242);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (324, 250, TO_DATE('2021-06-09 21:44:46', 'YYYY-MM-DD
HH24:MI:SS'), 261, 248);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (325, 700, TO_DATE('2021-06-09 21:47:19', 'YYYY-MM-DD
HH24:MI:SS'), 263, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (327, 250, TO_DATE('2021-06-09 21:48:40', 'YYYY-MM-DD
HH24:MI:SS'), 263, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (329, 200, TO_DATE('2021-06-09 21:49:27', 'YYYY-MM-DD
HH24:MI:SS'), 263, 243);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (331, 500, TO_DATE('2021-06-09 21:49:59', 'YYYY-MM-DD
HH24:MI:SS'), 263, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (333, 500, TO_DATE('2021-06-09 21:50:16', 'YYYY-MM-DD
HH24:MI:SS'), 263, 246);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (335, 300, TO_DATE('2021-06-09 21:50:32', 'YYYY-MM-DD
HH24:MI:SS'), 263, 246);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (337, 200, TO_DATE('2021-06-09 21:51:05', 'YYYY-MM-DD
HH24:MI:SS'), 263, 253);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (339, 250, TO_DATE('2021-06-09 21:51:52', 'YYYY-MM-DD
HH24:MI:SS'), 263, 244);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (340, 790, TO_DATE('2021-06-09 21:55:14', 'YYYY-MM-DD
HH24:MI:SS'), 265, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (341, 300, TO_DATE('2021-06-09 21:55:49', 'YYYY-MM-DD
HH24:MI:SS'), 265, 244);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (342, 400, TO_DATE('2021-06-09 21:55:56', 'YYYY-MM-DD
HH24:MI:SS'), 265, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (343, 900, TO_DATE('2021-06-09 21:56:04', 'YYYY-MM-DD
HH24:MI:SS'), 265, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (344, 300, TO_DATE('2021-06-09 21:56:27', 'YYYY-MM-DD
HH24:MI:SS'), 265, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (345, 410, TO_DATE('2021-06-09 21:57:08', 'YYYY-MM-DD
HH24:MI:SS'), 265, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (346, 200, TO_DATE('2021-06-09 21:57:15', 'YYYY-MM-DD
HH24:MI:SS'), 265, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (223, 1.23, TO_DATE('2021-06-08 12:27:15',
'YYYY-MM-DD HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (224, 2.34, TO_DATE('2021-06-08 12:30:22',
'YYYY-MM-DD HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (221, 100, TO_DATE('2021-06-08 11:22:42', 'YYYY-MM-DD
HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (222, 150, TO_DATE('2021-06-08 11:36:47', 'YYYY-MM-DD
HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (225, 3.45, TO_DATE('2021-06-08 12:46:16',
'YYYY-MM-DD HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (226, 7.77, TO_DATE('2021-06-08 12:58:37',
'YYYY-MM-DD HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (227, 0.01, TO_DATE('2021-06-08 13:01:18',
'YYYY-MM-DD HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (228, 10, TO_DATE('2021-06-08 13:01:39', 'YYYY-MM-DD
HH24:MI:SS'), 101, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
```



```
RESERVATION_TOURID) VALUES (267, 120, TO_DATE('2021-06-09 21:17:59', 'YYYY-MM-DD
HH24:MI:SS'), 255, 241);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (269, 1000.01, TO_DATE('2021-06-09 21:18:26',
'YYYY-MM-DD HH24:MI:SS'), 255, 254);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (271, 10.15, TO_DATE('2021-06-09 21:18:52',
'YYYY-MM-DD HH24:MI:SS'), 255, 254);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (273, 220, TO_DATE('2021-06-09 21:19:16', 'YYYY-MM-DD
HH24:MI:SS'), 255, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (275, 230, TO_DATE('2021-06-09 21:19:35', 'YYYY-MM-DD
HH24:MI:SS'), 255, 247);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (277, 1500, TO_DATE('2021-06-09 21:19:58',
'YYYY-MM-DD HH24:MI:SS'), 255, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (279, 200, TO_DATE('2021-06-09 21:20:14', 'YYYY-MM-DD
HH24:MI:SS'), 255, 243);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (281, 1000, TO_DATE('2021-06-09 21:20:37',
'YYYY-MM-DD HH24:MI:SS'), 255, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (283, 150.5, TO_DATE('2021-06-09 21:29:18',
'YYYY-MM-DD HH24:MI:SS'), 257, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (285, 1000, TO_DATE('2021-06-09 21:29:35',
'YYYY-MM-DD HH24:MI:SS'), 257, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (287, 400, TO_DATE('2021-06-09 21:30:04', 'YYYY-MM-DD
HH24:MI:SS'), 257, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (289, 150, TO_DATE('2021-06-09 21:31:29', 'YYYY-MM-DD
HH24:MI:SS'), 257, 246);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (291, 300, TO_DATE('2021-06-09 21:31:45', 'YYYY-MM-DD
HH24:MI:SS'), 257, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (293, 500, TO_DATE('2021-06-09 21:32:11', 'YYYY-MM-DD
HH24:MI:SS'), 257, 245);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (295, 299.5, TO_DATE('2021-06-09 21:32:34',
'YYYY-MM-DD HH24:MI:SS'), 257, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (297, 1000, TO_DATE('2021-06-09 21:32:59',
'YYYY-MM-DD HH24:MI:SS'), 257, 248);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (299, 150, TO_DATE('2021-06-09 21:33:15', 'YYYY-MM-DD
HH24:MI:SS'), 257, 244);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
```



```
RESERVATION_TOURID) VALUES (311, 700, TO_DATE('2021-06-09 21:42:38', 'YYYY-MM-DD
HH24:MI:SS'), 261, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (313, 210, TO_DATE('2021-06-09 21:42:53', 'YYYY-MM-DD
HH24:MI:SS'), 261, 161);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (315, 2, TO_DATE('2021-06-09 21:43:09', 'YYYY-MM-DD
HH24:MI:SS'), 261, 248);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (317, 700, TO_DATE('2021-06-09 21:43:36', 'YYYY-MM-DD
HH24:MI:SS'), 261, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (319, 300, TO_DATE('2021-06-09 21:43:50', 'YYYY-MM-DD
HH24:MI:SS'), 261, 242);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (321, 700, TO_DATE('2021-06-09 21:44:23', 'YYYY-MM-DD
HH24:MI:SS'), 261, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (323, 250, TO_DATE('2021-06-09 21:44:35', 'YYYY-MM-DD
HH24:MI:SS'), 261, 252);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (326, 400, TO_DATE('2021-06-09 21:48:30', 'YYYY-MM-DD
HH24:MI:SS'), 263, 253);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (328, 500, TO_DATE('2021-06-09 21:48:45', 'YYYY-MM-DD
HH24:MI:SS'), 263, 244);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (330, 600, TO_DATE('2021-06-09 21:49:39', 'YYYY-MM-DD
HH24:MI:SS'), 263, 242);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (332, 900, TO_DATE('2021-06-09 21:50:07', 'YYYY-MM-DD
HH24:MI:SS'), 263, 186);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (334, 200, TO_DATE('2021-06-09 21:50:25', 'YYYY-MM-DD
HH24:MI:SS'), 263, 254);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (336, 500, TO_DATE('2021-06-09 21:50:44', 'YYYY-MM-DD
HH24:MI:SS'), 263, 253);
INSERT INTO BD_PALCZAK.PAYMENT (ID, AMOUNT, PAYMENTDATE, RESERVATION_CUSTOMERID,
RESERVATION_TOURID) VALUES (338, 200, TO_DATE('2021-06-09 21:51:39', 'YYYY-MM-DD
HH24:MI:SS'), 263, 243);
```