1. Wprowadzenie

…………………………………………………………………..

1. Analiza wymagań

**Użytkownicy i ich role:**

1. **Administrator** (ALL)
   * Może wszystko
2. **Pracownik:**
3. REJESTRATOR (WRITE, REWRITE, DELETE)
   * Rejestracja konferencji.
4. KONTROLER (READ, REWRITE, DELETE)
   * zmiana statusów\*,
   * kontrolowanie: terminów konferencji, wpłat, uzupełnienia danych,
   * wysyłanie powiadomień *(przypomnienie o płatności, nieuzupełnione dane uczestników – klient zbiorowy),*
   * zwrot pieniędzy w przypadku rezygnacji uczestnika lub anulowania konferencji.
5. RAPORTER (READ)
   * generowanie list osobowych uczestników na konferencje i warsztaty,
   * podawanie informacji o płatnościach i informacji o stałych klientach.
6. OBSŁUGA - Obsługa konferencji (READ, REWRITE)
   * Drukowanie oraz rozdawanie identyfikatorów dla uczestników konferencji według listy, kontrola posiadania legitymacji studenckiej w przypadku studentów – regulacja wpłaty w przypadku braku.
7. **Klient:**
8. Klient indywidualny:
   * Rejestracja na konferencje i warsztaty (WRITE),
   * Aktualizacja i wyświetlanie - płatności, zmiana warsztatów, rezygnacja z udziału w konferencji/warsztacie (REWRITE, DELETE, READ).
9. Klient zbiorczy – Firma:
   * Rezerwacja miejsc na konferencje i warsztaty (WRITE),
   * Aktualizacja i wyświetlanie – płatności, zmiana rezerwacji, rezygnacja z rezerwacji (REWRITE, DELETE, READ),
   * Rejestracja uczestników (WRITE),
   * Aktualizacja i wyświetlanie - zmiana warsztatów, rezygnacja z konferencji/warsztatów (REWRITE, DELETE, READ).

**\*Statusy:**

1. **Statusy konferencji:**

* W trakcie – zapisy trwają,
* Zamknięty – zamknięcie rejestracji na konferencję z powodu zbyt małej liczby chętnych lub problemów technicznych,
* Zakończony – rejestracja na daną konferencję zakończona z powodu braku miejsc , zbliżającej się daty konferencji,
* Skompletowany – status ustawiany dla zakończonej konferencji.

1. **Statusy rejestracji:**

* W trakcie - możliwość aktualizacji, dokonania wpłaty, rezygnacji,
* Zakończona - dokonano wpłaty, możliwość rezygnacji z udziału,
* Anulowana – rezygnacja z udziału lub anulowanie spowodowane nie uiszczeniem opłaty.

1. **Statusy rezerwacji:**

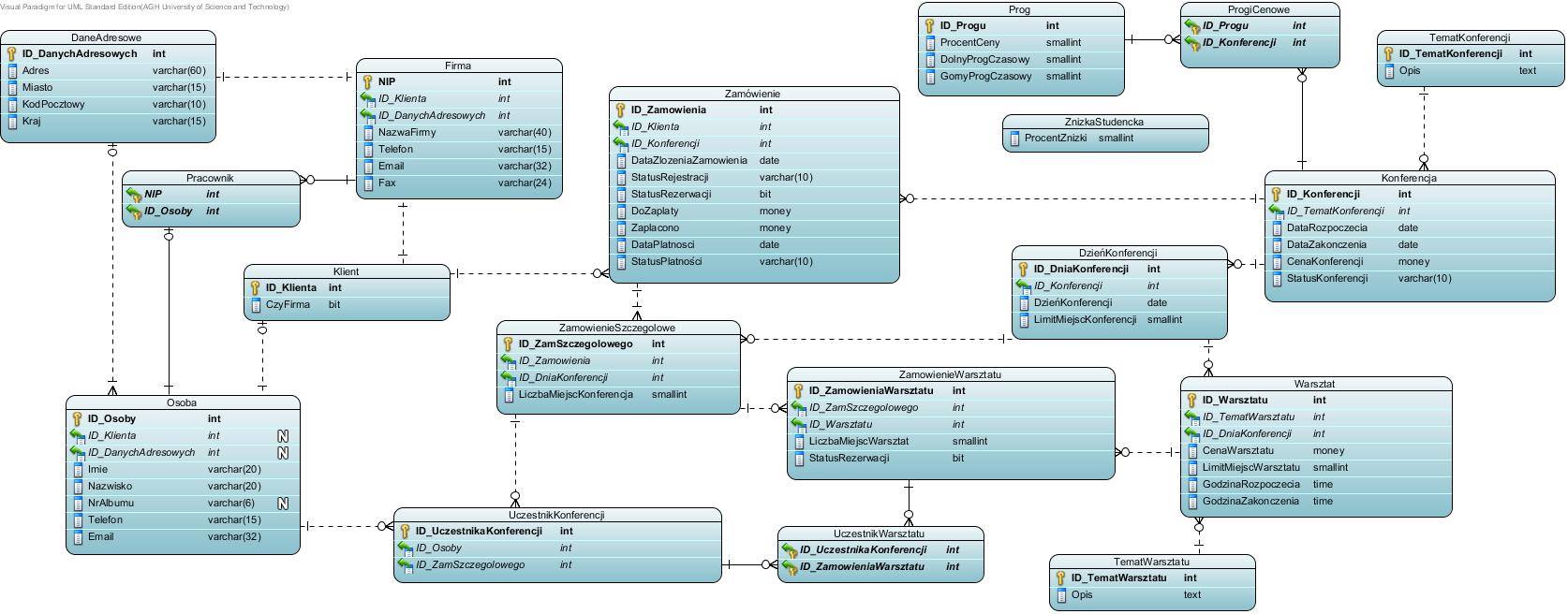
* W trakcie – nie podano danych uczestników,
* Zakończona – podano dane uczestników,

1. **Statusy płatności:**
   * Niezapłacone – uczestnik nadal ma możliwość zapłaty,
   * Zapłacone – uczestnik wpłacił pieniądze w terminie,
   * Zwrot – uczestnik wpłacił pieniądze ale anulował rezerwacje 2 tygodnie przed rozpoczęciem lub konferencja nie została zorganizowana z powodów losowych, uczestnik otrzymał 100% zwrot
   * Anulowane – uczestnik wpłacił pieniądze ale anulował rezerwacje zbyt późno by otrzymać zwrot.

**Proces rejestracji na konferencje i warsztaty:**

* 1. **Rejestracja na konferencję i warsztaty:**
     + Klient indywidualny – normalny przebieg rejestracji – podanie danych osobowych, kontaktowych, w przypadku bycia studentem należy dodatkowo podać nr legitymacji studenckiej, wybór konferencji i warsztatów,
     + Klient zbiorczy – może dokonać rezerwacji miejsc na konferencji i warsztatach, a do dwóch tygodni przed konferencją musi zarejestrować uczestników *(w przypadku nieuzupełnienia danych – powiadomienie ze strony firmy organizującej*),
     + W przypadku konferencji kilkudniowych można zarejestrować się na wybrane dni,
     + Ograniczona ilość miejsc na konferencje i warsztaty.
  2. **Rejestracja na warsztaty:**
* Warunek konieczny – rejestracja na konferencję w danym dniu,
* Można wybrać tylko jeden warsztat, który trwa w tym samym czasie.
  1. **Opłata za konferencje i warsztaty:**
* Czas na dokonanie wpłaty – tydzień od rejestracji,
* Cena konferencji zależy od wykupionych usług i terminu ich wykupienia – im bliżej terminu konferencji tym wyższa opłata,
* *Każda konferencja ma swoje procentowe progi cenowe zależne od czasu który pozostał do rozpoczęcia konferencji*
* Cena warsztatów - stała opłata lub jej brak (zależy wyłącznie od warsztatów),
* *Istnieje jedna 50% zniżka dla studentów, która zostaje naliczona od aktualnej ceny konferencji (po wyliczeniu z obowiązującego w danej chwili progu cenowego)*
* W przypadku nie uiszczenia opłaty – rejestracja zostaje anulowana; *uczestnik może poprosić o przedłużenie terminu wpłaty*
* *Jeśli uczestnik wpłaci nieodpowiednią kwotę otrzymuje powiadomienie od firmy*
* *Podczas odbioru identyfikatora na konferencje student jest zobowiązany okazać legitymacje studencką zadeklarowaną podczas rejestracji – w przeciwnym wypadku osoba musi dopłacić różnicę między zwykłą ceną a uregulowaną ceną zniżkową.*
* *W przypadku rezygnacji z konferencji/warsztatu do 2 tygodni przed rozpoczęciem możliwe jest otrzymanie 100% zwrotu wpłaconej kwoty, później nie jest to możliwe*

1. Schemat

******

1. Opis tabel

DaneAdresowe - tabela przechowująca dane adresowe (adres, miasto, kod pocztowy, kraj) wszystkich klientów (zarówno firm jak i klientów prywatnych).

Sprawdzane warunki: adres e-mail musi być w formacie %\_@\_%.\_%

Firma - tabela przechowująca informacje na temat klientów-firm. Klucz główny stanowi

niepowtarzalny 5-cio znakowy NIP, w tabeli znajduje się również nazwa firmy oraz klucze obce do

tabel DaneAdresowe i Klient.

Osoba - tabela przechowująca informacje o uczestnikach i klientach prywatnych firmy - imię, nazwisko, nr albumu, telefon, email. Jeśli osoba nie jest klientem to klucz obcy do tabeli Klient ma wartość null. Wszystkie wartości pola ID\_Klienta inne niż null muszą być unikalne.

Klient - tabela łącznikowa przechowująca ID\_Klienta oraz informację o tym czy jest on firmą w postaci bitowej. ID\_Klienta stanowi klucz obcy w tabelach Osoba i Firma.

Konferencja - tabela przechowująca informacje o wszystkich konferencjach organizowanych przez firmę, temat konferencji przechowywany w postaci klucza obcego do tabeli TematKonferencji. W polu StatusKonferencji znajduje się informacja, czy dana konferencja odbyła się już etc.

TematKonferencji - tabela przechowująca dostępne tematy konferencji. Id\_TematuKonferencji jest kluczem obcym w tabeli Konferencja.

Pracownik - tabela łącząca tabele Firma i Osoba – kluczem głównym jest połączenie NIPu i ID\_Osoby.

Dzień\_Konferencji – tabela przechowująca dane tj. data i limit miejsc w danym dniu. ID\_Konferencji stanowi klucz obcy.

Warsztat - tabela przechowująca informacje o wszystkich warsztatach organizowanych przez firmę w danym dniu konferencji (ID\_TematuWarsztatu – klucz obcy).

TematWarsztatu - tabela przechowująca dostępne tematy warsztatów. Id\_TematuWarsztatu jest kluczem obcym w tabeli Warsztat.

Zamówienie – tabela przechowująca informajce na temat wszystkich składanych zamówień (data

zamówienia, liczba zarezerwowanych miejsc, kwota do zapłaty, kwota wpłącona, etc.). W tabeli są przetrzymywane klucze obce do FormyPłatności, Klienta zamawiającego wycieczkę oraz Turnusu

którego dotyczy zamówienie

ZamówienieSzczegółowe – tabela łącząca zamówienie z poszczególnymi dniami konferencji – przechowuje klucze obce: ID\_DniaKonferencji I ID\_Zamówienia oraz liczbe zarezerwowanych miejsc

ZamówienieWarsztatu - tabela łącząca zamówienie szczegółowe z zamówionymi warsztatami

UczestnikKonferencji – tabela łącznikowa, przechowuje klucze obce Osoby i Zamówienia Szczegółowego

UczestnikWarsztatu – tabela łącznikowa, przechowuje klucze obce Uczestnika Konferencji i Zamówienia Warsztatu

1. Procedury i triggery

nowy\_adres

@Adres NVARCHAR(45), @KodPocztowy NVARCHAR(10), @Miasto NVARCHAR(25) , @Kraj NVARCHAR(25)

dodaj\_klienta\_osoba

@Imie NVARCHAR(20), @Nazwisko NVARCHAR(20), @NrAlbumu NVARCHAR(6), @Telefon NVARCHAR(25), @Email NVARCHAR(45), @Adres NVARCHAR(60), @Miasto NVARCHAR(15), @KodPocztowy NVARCHAR(10), @Kraj NVARCHAR(15)

dodaj\_osobe\_jako\_klienta

@ID\_Osoby INT

dodaj\_klienta\_firma

@NIP INT, @NazwaFirmy nvarchar(45), @Telefon NVARCHAR(25), @Fax NVARCHAR(24) , @Email NVARCHAR(45), @Adres NVARCHAR(60), @Miasto NVARCHAR(15), @KodPocztowy NVARCHAR(10), @Kraj NVARCHAR(15)

dodaj\_osobe

@Imie NVARCHAR(20), @Nazwisko NVARCHAR(20), @NrAlbumu NVARCHAR(6), @Telefon NVARCHAR(25) , @Email NVARCHAR(45)

dodaj\_pracownika

@NIP INT, @ID\_Osoby INT

dodaj\_konferencje

@ID\_TematuKonferencji INT, @DataRozpoczecia DATE, @DataZakonczenia DATE, @CenaKonferencji MONEY, @StatusKonferencji SMALLINT

dodaj\_temat\_konferencji

@Opis TEXT

dodaj\_warsztat

@ID\_TematuWarsztatu INT, @ID\_DniaKonferencji INT, @CenaWarsztatu MONEY, @LimitMiejscWarsztat SMALLINT, @GodzRozpoczecia TIME, @GodzinaZakonczenia TIME

dodaj\_temat\_warsztatu

@Opis TEXT

dodaj\_dzien\_konferencji

@ID\_Konferencji INT, @DzienKonferencji DATE, @LimitMiejscKonferencja SMALLINT

dodaj\_zamowienie

@ID\_Klienta INT, @ID\_Konferencji INT, @DataZlozeniaZamowienia DATE, @StatusRejestracji NVARCHAR(10), @StatusRezerwacji BIT, @DoZapltay MONEY, @Zaplacono MONEY, @TerminPlatnosci DATE, @StatusPlatnosci NVARCHAR(10)

dodaj\_zamowienie\_szcz

@ID\_Zamowienia INT, @ID\_DniaKonferencji INT, @LiczbaMiejscKonferencja SMALLINT

dodaj\_zamowienie\_warsztatu

@ID\_ZamSzczegolowego INT, @ID\_Warsztatu INT, @LiczbaMiejscWarsztat SMALLINT, @StatusRezerwacji BIT

dodaj\_uczestnika\_konferencji

@ID\_Osoby INT, @ID\_ZamowieniaSzczegolowego INT

dodaj\_uczestnika\_warsztatu

@ID\_UczestnikaKonferencji INT, @ID\_ZamowieniaWarsztatu INT

1. Widoki
2. Generator danych

Generator danych jest programem napisanym w języku Python. Dane są generowane na podstawie plików :

- im\_z.txt, imiona\_m.txt - imiona odpowiednio żeńskie i męskie

- nazw.txt - nazwiska dla klientów i uczestników

- firmy.txt - nazwy firm klientów

- miasta.txt - nazwy miast i kody pocztowe

- ulice.txt - nazwy ulic wraz z numerami

- tematy\_k.txt , tematy\_w.txt – tematy warsztatów

Na podstawie tych plików i parametrów wejściowych programu zostaje wygenerowany plik sql z wywołaniami odpowiednich procedur uzupełniających poszczególne tabele.

4. Procedury i warunki integralnościowe

Procedury:

nowe\_dane\_teleadresowe

@Adres nvarchar(45) - dokładny adres (ulica, numer lokalu)

@KodPocztowy nvarchar(10)

@Miasto nvarchar(25)

@Kraj nvarchar(25)

@Telefon varchar(25) [nieobowiązkowy]

@Fax varchar(25) [nieobowiązkkowy]

@Email varchar(45) [nieobowiązkowy] - adres e-mail w formacie \*@\*.\*

Procedura dodaje nowy wpis w tabeli DaneTeleadresowe, jest wykorzystywana przez procedury

dodające osoby, hotele oraz firmy.

dodaj\_firme

@NIP varchar(20)

@NazwaFirmy nvarchar(20)

@Adres nvarchar(45) - dokładny adres (ulica, numer lokalu)

@KodPocztowy nvarchar(10)

@Miasto nvarchar(25)

@Kraj nvarchar(25)

@Telefon varchar(25) [nieobowiązkowy]

@Fax varchar(25) [nieobowiązkowy]

@Email varchar(45) [nieobowiązkowy] - adres e-mail w formacie \*@\*.\*

Procedura pozwala dodać nową firmę - klienta. Dodaje ona odpowiednie wpisy w tabelach

DaneTeleadresowe, Klienci, Firmy. Jeśli którakolwiek z tych tabel nie będzie mogła zostać zmieniona

(np. z powodu błędnych danych), żadna akcja nie zostanie wykonana.

dodaj\_hotel

@Nazwa nvarchar(30)

@Adres nvarchar(45) - dokładny adres (ulica, numer lokalu)

@KodPocztowy nvarchar(10)

@Miasto nvarchar(25)

@Kraj nvarchar(25)

@Telefon varchar(25) [nieobowiązkowy]

@Fax varchar(25) [nieobowiązkowy]

@Email varchar(45) [nieobowiązkowy] - adres e-mail w formacie \*@\*.\*

Procedura pozwala dodać nowy hotel, w którym będą kwaterowani uczestnicy wycieczek. Hotel

domyślnie oznaczany jest jako aktywny. Procedura dodaje odpowiednie wpisy w tabelach: Hotele,

DaneTeleadresowe. Jeśli którakolwiek z tych tabel nie będzie mogła zostać zmieniona (np. z powodu

błędnych danych), żadna akcja nie zostanie wykonana.

dodaj\_osobe\_kli

@Imie nvarchar(45)

@DrugieImie nvarchar(45)

@DataUrodzenia date - data urodzenia, przyjmowany format to ‘dd/mm/yyyy’

@Nazwisko nvarchar(45)

@Plec char(1) - ‘K’ - kobieta, ‘M’ - mężczyzna

@Adres nvarchar(45) - dokładny adres (ulica, numer lokalu)

@KodPocztowy nvarchar(10)

@Miasto nvarchar(25)

@Kraj nvarchar(25)

@Telefon varchar(25) [nieobowiązkowe]

@Fax varchar(25) [nieobowiązkowe]

@Email varchar(45) [nieobowiązkowe] - adres e-mail w formacie \*@\*.\*

Procedura dodająca do bazy danych nową osobę, przy wykupowaniu przez nią wycieczki. Dodawane

są odpowiednie wpisy w tabelach: Osoby, Klienci, DaneTeleadresowe. Jeśli którakolwiek z tych tabel

nie będzie mogła zostać zmieniona (np. z powodu błędnych danych), żadna akcja nie zostanie

wykonana.

Uwaga: jeśli dane osoby są już w bazie danych (była wcześniej uczestnikiem wycieczki), nie należy

używać tej procedury. Jeśli chcemy dodać osobę, która nie jest klientem, należy użyć procedury

dodaj\_osobe.

dodaj\_osobe

@Imie nvarchar(45)

@DrugieImie nvarchar(45)

@Nazwisko nvarchar(45)

@DataUrodzenia date - data urodzenia, przyjmowany format to ‘dd/mm/yyyy’

@Plec char(1) - ‘K’ - kobieta, ‘M’ - mężczyzna

@Adres nvarchar(45)

@KodPocztowy nvarchar(10)

@Miasto nvarchar(25)

@Kraj nvarchar(25)

@Telefon varchar(25) [nieobowiązkowe]

@Fax varchar(25) [nieobowiązkowe]

@Email varchar(45) [nieobowiązkowe] - adres e-mail w formacie \*@\*.\*

Procedura dodająca do bazy danych nową osobę, gdy nie jest ona klientem (jest tylko uczestnikiem

wycieczki). Dodawane są odpowiednie wpisy w tabelach: Osoby, Klienci, DaneTeleadresowe. Jeśli

którakolwiek z tych tabel nie będzie mogła zostać zmieniona (np. z powodu błędnych danych), żadna

akcja nie zostanie wykonana.

dodaj\_osobe\_jako\_klienta

@IDOsoby int - identyfikator osoby, która została klientem

Procedura dodająca osobę niebędącą klientem do bazy klientów. Zwraca błąd, jeśli osoba o danym

identyfikatorze jest już klientem. W przypadku nowego klienta zamiennie można używać procedury

dodaj\_osobe\_kli oraz kombinacji procedur dodaj\_osobe i dodaj\_osobe\_jako\_klienta, jednakże jeśli

osoba była już wcześniej zarejestrowana w bazie danych (ponieważ była uczestnikiem wycieczki)

należy użyć powyższej procedury.

dolacz\_atrakcje

@IDRodzajeWycieczek int - identyfikator wycieczki, do której dodajemy nową możliwą

atrakcję

@IDRodzajeAtrakcji int - identyfikator atrakcji, którą dodajemy

Procedura pozwala na połączenie istniejącego rodzaju wycieczki z rodzajem atrakcji. Połączenie to

oznacza, że uczestnik biorący udział w danym rodzaju wycieczki może dodatkowo zapisać się na dany

rodzaj atrakcji (jeśli w turnusie są jeszcze miesjca).

dodaj\_uczestnika\_wycieczki

@idZamowienia int - identyfikator zamówienia na wycieczkę, w której weźmie udział dana

osoba

@idOsoby int - identyfikator osoby zapisującej się na wycieczkę

Procedura pozwala na zapisanie osoby na wycieczkę, która jest zamawiana (rezerwowana). Nie

można zapisać na wycieczkę więcej osób, niż jest zarezerwowanych miejsc w odpowiednim

zamówieniu.

dodaj\_uczestnika\_atrakcji

@idZamowienia int - identyfikator zamówienia, podczas realizacji którego wczasowicz

weźmie udział w dodatkowej atrakcji

@idRodzajeAtrakcji int - identyfikator atrakcji, w której wczasowicz zamierza wziąć udział

@idOsoby int - identyfikator osoby zapisującej się na dodatkową atrakcję

Procedura pozwala na zapisanie uczestnika wycieczki na dodatkową atrakcję. Oczywistym jest, że

osoba o danym ID musi być uprzednio zapisana na wycieczkę, na której dana atrakcja jest dostępna.

Należy też wziąć pod uwagę, że każda atrakcja ma górny limit miejsc, który nie może zostać

przekroczony.

deaktywuj\_hotel

@idHotelu int - identyfikator deaktywowanego hotelu

Procedura oznacza hotel jako nieaktywny. Oznacza to, że hotel już nie istnieje bądź nie prowadzimy z

nim współpracy.

deaktywuj\_wycieczke

@idRodzajeWycieczek int - identyfikator deaktywowanej wycieczki

Procedura oznacza wycieczkę jako nieaktywną. Oznacza to, że nie organizujemy już nowych turnusów

na tego rodzaju wycieczki, jednak zamówienia złożone zostaną zreazlizowane.

deaktywuj\_atrakcje

@idRodzajeAtrakcji int - identyfikator deaktywowanej atrakcji

Procedura oznacza atrakcję jako nieaktywną. Oznacza to, że nie można już się na nią zapisywać.

Zapisane osoby wciąż mogą jednak na nią jechać.

reaktywuj\_hotel

@idHotelu int - identyfikator deaktywowanego hotelu

Procedura oznacza hotel jako aktywny. Oznacza to, że hotel znów współpracuje z naszą firmą.

deaktywuj\_wycieczke

@idRodzajeWycieczek int - identyfikator deaktywowanej wycieczki

Procedura oznacza wycieczkę jako aktywną. Oznacza to, że są aktualnie organizowane nowe turnusy

na ten rodzaj wycieczki.

deaktywuj\_atrakcje

@idRodzajeAtrakcji int - identyfikator deaktywowanej atrakcji

Procedura oznacza atrakcję jako aktywną. Oznacza to, że można się zapisywać na ten rodzaj atrakcji.

usun\_uczestnika\_wycieczki

@idOsoby int - identyfikator osoby, która rezygnuje z uczestnictwa w wycieczcce

@idZamowienia int - identyfikator zamówienia, z uczestnictwa w realizacji którego dana

osoba rezygnuje

Procedura pozwala wypisać osobę z uczestnictwa w wycieczcce. Jej dane pozostają w bazie, jednakże

usuwane jest powiązanie pomiędzy osobą a konkretnym zamówieniem. W jej miejsce należy zapisać

inną osobę, by całkowita liczba zapisanych osób zgadzała się z liczbą zarerezerwowanych miejsc.

usun\_uczestnika\_atrakcji

@idZamowienia int - identyfikator zamówienia/rezerwacji, w którym dana osoba jest

zapisana na atrakcję dodatkową

@idRodzajeAtrakcji int - identyfikator rodzaju atrakcji, na który jest zapisana rezygnująca

osoba

@idOsoby int - identyfikator osoby, która rezygnuje z uczestnictwa w atrakcji dodatkowej

Procedura pozwala wypisać osobę z uczestnictwa w danej atrakcji.

anuluj\_nieoplacone\_rezerwacje\_po\_terminie

(procedura bezparametrowa)

Ustawia stan wszystkich zamówień, w których nie opłacono pełnej kwoty po upływie 14 dni od

rezerwacji na ‘anulowano’ z powodem ‘Nie oplacono zamowienia przez 14 dni lub zamowienie nie

zostalo oplacone 7 dni przed rozpoczeciem turnusu’. Wcześniej dobrze jest skorzystać z

odpowiedniego widoku, by poinformować o tym fakcie klientów.

anuluj\_wszystkie\_zamowienia

@IDRodzajeWycieczek int

@IDTurnusy int

@IDHotelu int

@PowodAnulowania nvarchar(300)

Procedury należy używać z rozwagą! Powoduje ona anulowanie wszystkich zamówień spełniających

dany warunek. Należy podać jeden z trzech identyfikatorów, natomiast resztę uzupełnić słowem null.

Procedura wyszuka wszystkie nieanulowane, aktualne zamówienia związane z danym

turnusem/rodzajem wycieczki/hotelem i zmieni ich stan na anulowano (z podanym powodem). W

przypadku, jeśli więcej niż jeden identyfikator zostanie podany, procedura potraktuje te warunki jako

połączone spójnikiem LUB.

dodaj\_atrakcje

@Nazwa nvarchar(45) - nazwa atrakcji, którą dodajemy

@LiczbaMiejsc smallint - liczba miejsc jaką przewidujemy na daną atrakcję

@CenaTransportu money - cena transportu

@CenaZaOsobe money - cena za osobę

Procedura dodająca atrakcję o podanych paramatrach do bazy danych. Jeśli atrakcja o podanej

nazwie znajduje się już w bazie zwracany jest błąd wraz z komunikatem.

dodaj\_wycieczki

@IDHotelu int - identyfikator hotelu, w którym będą kwaterowani uczestnicy

@Nazwa nvarchar(45) - krótka nazwa wycieczki

@CenaZaOsobe money - cena zależna od liczby osób

@CenaTransportu money - cena transportu, niezależna od liczby osób

@Opis nvarchar(400) - szczegółowy opis wycieczki

Procedura dodająca wycieczkę o podanych parametrach do bazy danych. Wycieczka musi być

przyporzadkowana do konkretnego hotelu, dlatego nalezy też podać id hotelu. Domyślnie stan każdej

nowo dodanej wycieczki jest ustawiany na aktywny. Jeśli hotel w którym ma być zakwaterowana

wycieczka nie istnieje zwracany jest błąd wraz z komunikatem.

dodaj\_platnosc

@Nazwa nvarchar(45)

Procedura dodająca do bazy danych płatność o wskazanej nazwie. Jeśli podana płatność już istnieje

zwracany jest błąd z odpowiednim komunikatem.

rezerwuj\_wycieczke

@IDKlienta int - identyfikator klienta, który rezerwuje i płaci za wycieczkę

@IDFormyPlatnosci int - identyfikator formy płatności

@IDTurnusy int - identyfikator turnusu, na który klient rezerwuje miejsca

@LiczbaOsob int - liczba miejsc, które klient rezerwuje

@Rabat real - rabat udzielony klientowi na to zamówienie

@Zaliczka money - zaliczka wpłacona od razu przy rezerwacji

Procedura rezerwująca wycieczkę na podstawie IDKlienta. Klient podaje IDFormyPlatnosci wybierając

w ten sposób formę w jakiej chce dokonać zapłaty, IDTurnusu na który składa zamówienie,

LiczbęOsób dla jakiej ma być rezerwacja oraz zaliczkę jaką wpłaca. DataZamówienia w tabeli

Zamówienia zostaje uzupełniona przez aktualną datę tj. datę rezerwacji.

dodaj\_turnus

@IDRodzajeWycieczek int

@IloscMiejsc smallint

@DataStartu date

@DataKonca date

Procedura dodająca turnus dla konkretnej wycieczki. Wymaga sprecyzowania IDRodzajuWycieczek,

Ilości miejsc, daty startu oraz daty konca. W przypadku gdy data końca jest datą wcześniejsza niż data

startu lub nie istnieje wycieczka do której chcemy dodać turnus, zwracany jest błąd.

Wystaw\_Fakture

@IDZamowienia int

Procedura generuje fakturę dla danego zamówienia zawierającą dokładne wyliczenie jego kosztów.

Faktura wypisywana jest procedurą PRINT.

Funkcje

koszt\_zamowienia (@IDZamowienia int - identyfikator zamówienia)

Zwraca typ money - całkowity koszt zamowienia o danym identyfikatorze

Funkcja wylicza całkowity koszt zamówienia. Bierze pod uwagę:

- koszt wycieczki, czyli koszt transportu + liczba osób \* koszt na osobę;

- koszt atrakcji, czyli koszt transportu każdej atrakcji, na którą chociaż jedna osoba jest zapisana

+liczba osób zapisanych na atrakcję \* koszt atrakcji na osobę

liczba\_wolnych\_miejsc\_na\_turnus(@IDTurnusy int - identyfikator turnusu)

Zwraca typ smallint - liczba wolnych miejsc na podany turnus

Funkcja wylicza liczbę wolnych miejsc na dany turnus. Bierze pod uwagę liczbę zarezerwowanych

miejsc we wszystkich zamówieniach związanych z tym turnusem . Przy spójnej bazie danych,

zwracana liczba powinna być zawsze większa od zera.

liczba\_wolnych\_miejsc\_atrakcji(@IDTurnusy int - identyfikator turnusu ,

@IDRodzajeAtrakcji int - identyfikator atrakcji)

Zwraca typ smallint - liczba wolnych miejsc na daną atrakcję w tym turnusie

Funkcja wylicza liczbę wolnych miejsc na daną atrakcję w danym turnusie. Bierze pod uwagę liczbę

osób zapisanych na daną atrakcję ze wszystkich zamówień związanych z tym turnusem. Przy spójnej

bazie danych, zwracana liczba powinna być zawsze większa od zera.

Triggery:

LimitRezerwacji w tabeli Zamówienia

Dla zdarzen INSERT, UPDATE

Trigger pilnujący aby klient nie złożył zamówienia na liczbę osób przerastającą liczbę wolnych miejsc

dla danego turnusu.

UnikalnoscOsobyKlienta w tabeli Osoby

Dla zdarzeń INSERT, UPDATE

Uwaga: ze względów wydajnościowych można zmienić tylko jeden identyfikator klienta w jednym

zdarzeniu UPDATE.

Sprawdza, czy ID klienta jest unikalne. Zastosowanie triggera zamiast słowa Unique wynika z faktu, że

osoba może mieć w tym polu wartosć NULL, a słowo Unique traktuje dwie wartości NULL jako równe

sobie.

DodawanieUczestnikaAtrakcji w tabeli UczestnicyAtrakcji

Dla zdarzeń INSERT, UPDATE

Sprawdza, czy w danym turnusie jest jeszcze miejsce na daną atrakcję. Sprawdza też, czy osoba jest

zapisana na wycieczkę, która daną atrakcję dodatkową oferuje.

WartoscRabatu w tabeli Zamówienia

Dla zdarzeń INSERT, UPDATE

Sprawdza czy zmodyfikowany rabat na pewno należy do zbioru [0, 1]

OstatniaDataZamowienia w tabeli Zamowienia

Dla zdarzeń INSERT, UPDATE

Trigger pilnujący aby data wprowadzonego zamówienie poprzedzała datę startu wycieczki o co

najmniej 7 dni.

ZmianaLiczbyUczestnikow w tabeli Zamowienia

Dla zdarzenia UPDATE

Trigger pilnujący aby przy zmianie liczby uczestników danej wycieczki nie zejść poniżej liczby

uczestników już wprowadzonych do bazy danych.

NowyUczestnikWycieczki w tabeli UczestnicyWycieczki

Dla zdarzenia INSERT

Pilnuje, aby liczba uczestników wycieczki przypisanych do danego zamówienia nie przekroczyła liczby

zarezerwowanych w tym zamówieniu miejsc.

Widoki:

dbo.DostepneWycieczki - widok wszystkich dostępnych wycieczek dla klienta, data startu wycieczki

nie przekracza aktualnej daty

dbo.WidokKlientów - wyświetla szczegółowe informacje o wszystkich klientach biura podróży

(zarówno firmach jak i klientach prywatnych)

dbo.WidokUczestnikówWycieczek - wyświetla szczegółowe informacje o wszystkich uczestnikach

wycieczek

dbo.WidokUczestnikówAtrakcji - wyświetla szczegółowe informacje o wszystkich uczestnikach

atrakcji

dbo. NiezaplaconePoTerminie - wyświetla informacje o nieopłaconych zamówieniach,

>14 dni po zarezerwowaniu lub <7 dni przed początkiem turnusu

dbo. NiewpisaniUczestnicy - wyświetla informacje o zamówieniach, w których nie uzupełniono

jeszcze listy osób biorących udział w wycieczcce

(liczba zarezerwowanych miejsc jest mniejsza niż liczba wpisanych uczestników)

Role:

W celu zapewnienia bezpieczeństwa w bazie danych powinny zostać zdefiniowane następujące role:

- administrator

- właściciel firmy / dyrektor

- sprzedawca

- klient (dane dostępne publicznie)

5. Opis generatora danych