
Bazy danych

Autorzy: Czernecki Paweł, Dziarkowski Michał, Matuszyński Wojciech, Szkarłat Szymon

1. Wymagania i funkcje systemu

Role:

Role działają na zasadzie sum zezwoleń. Jeżeli podmiot jest zarówno uczestnikiem jak i klientem wycieczki, zyskuje uprawnienia związane z obiema rolami. W wypadku, gdy podmiot jest jednocześnie klientem biura i jego administratorem, przewidziane są dwa osobne konta (jedno na stronie biura, drugie na panelu administracyjnym)

Właściciel

- pełne uprawnienia (zmiana ról w systemie, usuwanie użytkowników, dostęp do danych uczestników i klientów)

Admin

- posiada konto w panelu administracyjnym
- może tworzyć/usuwać nowe wycieczki i zmieniać ich właściwości, w wypadku uzyskania odpowiedniego komunikatu od sekretarza/sekretarki.
- ma dostęp do listy uczestników i listy klientów dla każdej wycieczki/fakultetu. Może usunąć klienta z listy klientów wycieczki, jeżeli nie dopełni swych obowiązków.

Sekretarz/Sekretarka

- Odpowiada za płatności przebiegające między klientami a systemem
- W przypadku rezygnacji z wycieczki/fakultetu bądź niespełnienia wymogów, zwraca pieniądze klientom.
- Kontaktuje się z firmami przewoźniczymi i hotelarskimi w celu uzgodnienia zakwaterowania uczestników, a także dat i miejsc zbiórek
 - Tydzień przed rozpoczęciem wycieczki wysyła administratorom listę klientów, którzy dopełnili obowiązków
 - Informuje administratorów na temat zmiany/dodania szczegółów wycieczki (np. zmiana daty rozpoczęcia, udostępnienie miejsca i czasu zbiórki na wycieczkę)

Klient

- może być podmiotem indywidualnym, lub firmą
- posiada konto na stronie
- może rezerwować wycieczki/fakultety
- może rezygnować z wycieczki/fakultetu
- może podawać dane uczestników
- ma dostęp do listy zarezerwowanych wycieczek i swoich uczestników, które na nią idą

Uczestnik

- posiada konto na stronie
- może zobaczyć wycieczki i fakultety, na które jest zapisany. Dla każdej wycieczki/fakultetu może sprawdzić miejsce i czas zbiórki(np. czas wylotu i lokalizacja lotniska na wycieczkę zagranicą), a dla wycieczek może też sprawdzić miejsce i czas przybycia na miejsce, oraz datę powrotu.

Działanie biura

Wycieczki

Główna część oferty biura podróży, składa się z jedno- i wielodniowych wypraw, zakup takiej oferty wiąże się z przygotowaniem dojazdu i zakwaterowania dla uczestników takowej wycieczki. Jest warunkiem koniecznym, do zakupu związanego fakultetu. Każda wycieczka ma własny dla siebie limit miejsc.

Fakultety

Poboczna część oferty biura. Klient może urozmaicić wycieczkę uczestników poprzez zakup usług dodatkowych, które odbędą się w czasie wycieczki. Aby uczestnik mógł wziąć udział w fakultecie, musi być uczestnikiem związanej wycieczki. Każdy fakultet ma własny dla siebie limit miejsc.

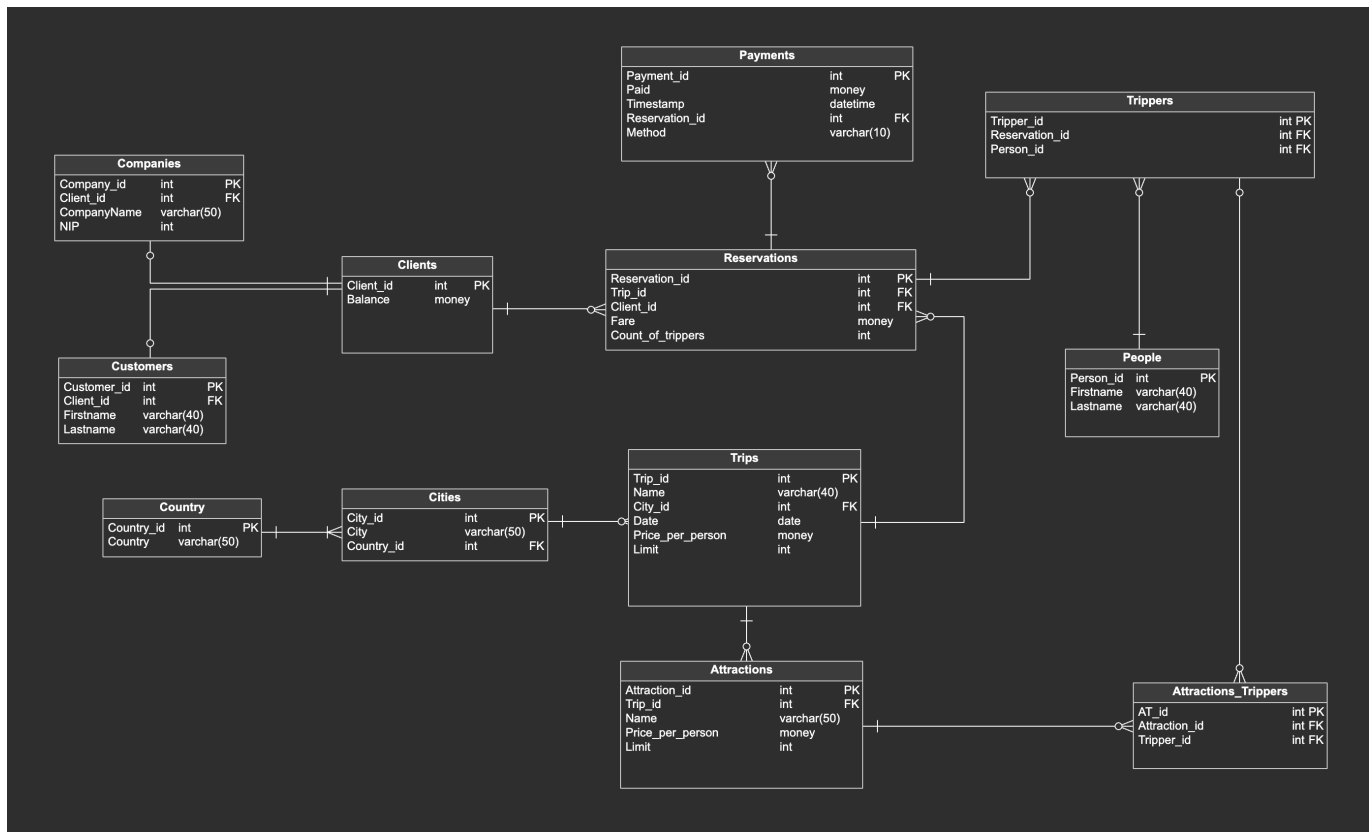
Rezerwacje Ofert

Klienci rezerwują wycieczki i fakultety w imieniu jednego lub wielu uczestników. Przy rezerwacji wycieczki podają liczbę uczestników. W terminie do tygodnia przed rozpoczęciem wycieczki klienci są zobowiązani do podania listy uczestników(Dane kontaktowe jak Imię i Nazwisko), oraz wpłacenia pełnej kwoty za wycieczkę. W przypadku niedopełnienia obowiązku, rezerwacja jest usuwana przez administratora, a wpłacone dotychczas pieniądze zwracane są przez sekretarkę. Fakultet można zamówić w dowolnym momencie po zamówieniu wycieczki, do tygodnia przed datą rozpoczęcia wycieczki. Musi przypisać do niego uczestnika imiennie do tygodnia przed rozpoczęciem wycieczki W przypadku, kiedy klient zmniejszy liczbę uczestników biorących udział w wycieczce/fakultecie, przysługuje mu zwrot wpłaconych pieniędzy pod warunkiem, że poinformował o tym fakcie biuro na tydzień przed rozpoczęciem wycieczki.

Zbiórki

Informacje na temat lokalizacji i daty, gdzie mają zebrać się uczestnicy w celu rozpoczęcia wycieczki dostarczane są nie później niż tydzień przed rozpoczęciem wycieczki. Data zbiórki wycieczki może być jednorazowo przesunięta do 24 godzin w przyszłość, nie później niż dobę przed oryginalną datą zbiórki. Informacje na temat zbiórki na usługi fakultatywne pojawiają się nie później niż dobę przed datą rozpoczęcia usługi fakultatywnej

2. Schemat bazy danych



3. Zapytania SQL tworzące tabele

Tworzenie tabel

```

CREATE TABLE Attractions (
    Attraction_id int NOT NULL,
    Trip_id int NOT NULL,
    Name varchar(50) NOT NULL,
    Price_per_person money NOT NULL,
    Limit int NOT NULL,
    CONSTRAINT check_1 CHECK ({Price_per_person > 0} {Limit > 0}),
    CONSTRAINT Attractions_pk PRIMARY KEY (Attraction_id)
);

CREATE TABLE Attractions_Trippers (
    AT_id int NOT NULL,
    Attraction_id int NOT NULL,
    Tripper_id int NOT NULL,
    CONSTRAINT Attractions_Trippers_pk PRIMARY KEY (AT_id)
);
  
```

```
CREATE TABLE Cities (  
    City_id int NOT NULL,  
    City varchar(50) NOT NULL,  
    Country_id int NOT NULL,  
    CONSTRAINT Cities_pk PRIMARY KEY (City_id)  
);  
  
CREATE TABLE Clients (  
    Client_id int NOT NULL,  
    Balance money NOT NULL,  
    CONSTRAINT Clients_pk PRIMARY KEY (Client_id)  
);  
  
CREATE TABLE Companies (  
    Company_id int NOT NULL,  
    Client_id int NOT NULL,  
    CompanyName varchar(50) NOT NULL,  
    NIP int NOT NULL,  
    CONSTRAINT Companies_pk PRIMARY KEY (Company_id)  
);  
  
CREATE TABLE Country (  
    Country_id int NOT NULL,  
    Country varchar(50) NOT NULL,  
    CONSTRAINT Country_pk PRIMARY KEY (Country_id)  
);  
  
CREATE TABLE Customers (  
    Customer_id int NOT NULL,  
    Client_id int NOT NULL,  
    Firstname varchar(40) NOT NULL,  
    Lastname varchar(40) NOT NULL,  
    CONSTRAINT Customers_pk PRIMARY KEY (Customer_id)  
);  
  
CREATE TABLE Payments (  
    Payment_id int NOT NULL,  
    Paid money NOT NULL,  
    Timestamp datetime NOT NULL,  
    Reservation_id int NOT NULL,  
    Method varchar(10) NOT NULL,  
    CONSTRAINT Payments_pk PRIMARY KEY (Payment_id)  
);  
  
CREATE TABLE People (  
    Person_id int NOT NULL IDENTITY(1, 1),  
    Firstname varchar(40) NOT NULL,  
    Lastname varchar(40) NOT NULL,  
    CONSTRAINT People_pk PRIMARY KEY (Person_id)  
);  
  
CREATE TABLE Reservations (  
    Reservation_id int NOT NULL,  
    Trip_id int NOT NULL,
```

```
Client_id int NOT NULL,  
Fare money NOT NULL,  
Count_of_trippers int NOT NULL,  
CONSTRAINT Reservations_pk PRIMARY KEY (Reservation_id)  
);  
  
CREATE TABLE Trippers (  
    Tripper_id int NOT NULL,  
    Reservation_id int NOT NULL,  
    Person_id int NOT NULL,  
    CONSTRAINT Trippers_pk PRIMARY KEY (Tripper_id)  
);  
  
CREATE TABLE Trips (  
    Trip_id int NOT NULL,  
    Name varchar(40) NOT NULL,  
    City_id int NOT NULL,  
    Date date NOT NULL,  
    Price_per_person money NOT NULL,  
    Limit int NOT NULL,  
    CONSTRAINT check_1 CHECK ({Price_per_person > 0}),  
    CONSTRAINT Trips_pk PRIMARY KEY (Trip_id)  
);
```

Relacje pomiędzy tabelami

```
ALTER TABLE Attractions_Trippers ADD CONSTRAINT Attraction_Tripper_Attractions  
    FOREIGN KEY (Attraction_id)  
    REFERENCES Attractions (Attraction_id);  
  
ALTER TABLE Attractions_Trippers ADD CONSTRAINT Attractions_Trippers_Trippers  
    FOREIGN KEY (Tripper_id)  
    REFERENCES Trippers (Tripper_id);  
  
ALTER TABLE Cities ADD CONSTRAINT Cities_Country  
    FOREIGN KEY (Country_id)  
    REFERENCES Country (Country_id);  
  
ALTER TABLE Companies ADD CONSTRAINT Companies_Clients  
    FOREIGN KEY (Client_id)  
    REFERENCES Clients (Client_id);  
  
ALTER TABLE Customers ADD CONSTRAINT Customers_Clients  
    FOREIGN KEY (Client_id)  
    REFERENCES Clients (Client_id);  
  
ALTER TABLE Payments ADD CONSTRAINT Payments_Reservations  
    FOREIGN KEY (Reservation_id)  
    REFERENCES Reservations (Reservation_id);
```

```
ALTER TABLE Reservations ADD CONSTRAINT Reservations_Clients
    FOREIGN KEY (Client_id)
    REFERENCES Clients (Client_id);

ALTER TABLE Reservations ADD CONSTRAINT Reservations_Trips
    FOREIGN KEY (Trip_id)
    REFERENCES Trips (Trip_id);

ALTER TABLE Trippers ADD CONSTRAINT Trippers_People
    FOREIGN KEY (Person_id)
    REFERENCES People (Person_id);

ALTER TABLE Trippers ADD CONSTRAINT Trippers_Reservations
    FOREIGN KEY (Reservation_id)
    REFERENCES Reservations (Reservation_id);

ALTER TABLE Attractions ADD CONSTRAINT Trips_Attractions
    FOREIGN KEY (Trip_id)
    REFERENCES Trips (Trip_id);

ALTER TABLE Trips ADD CONSTRAINT Trips_Cities
    FOREIGN KEY (City_id)
    REFERENCES Cities (City_id);
```