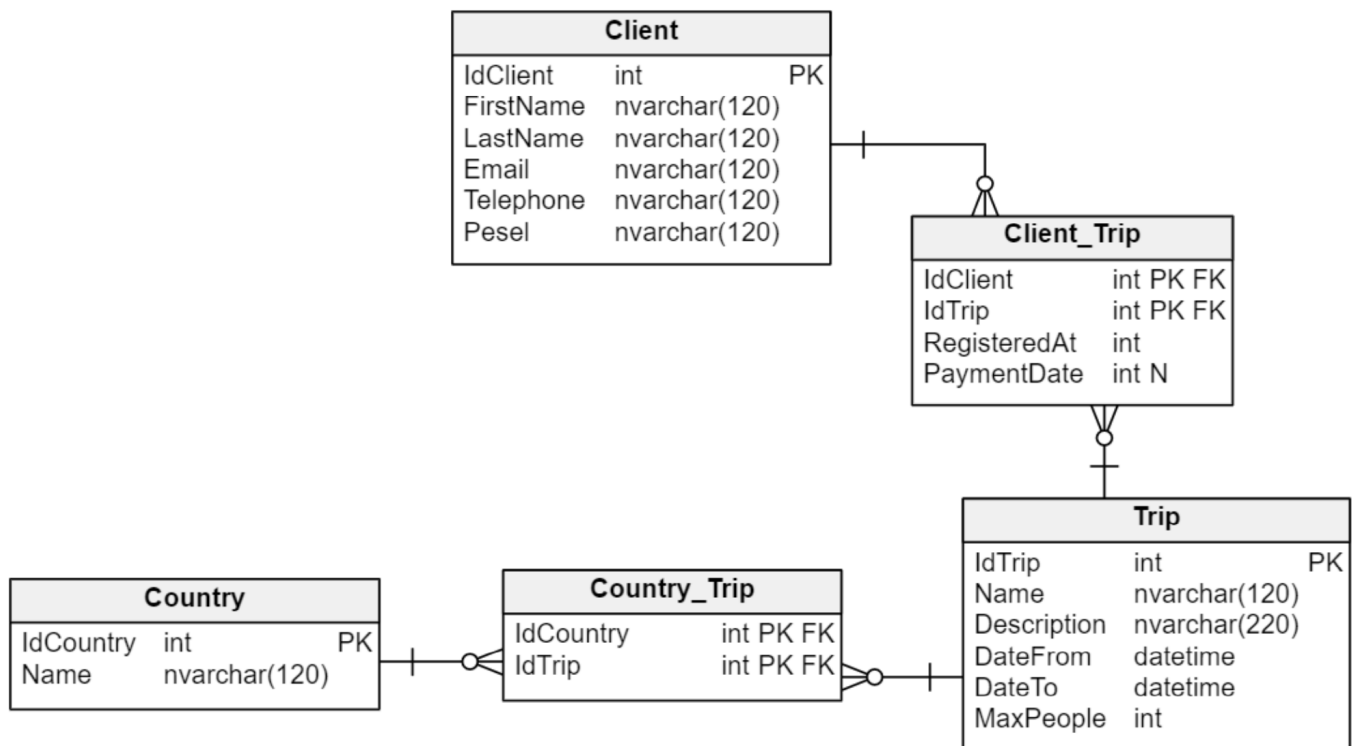


W niniejszym zadaniu skorzystamy z biblioteki EntityFramework Core - podejście "database first". Poniżej przedstawiony jest diagram na którym będziemy pracować.



Wszystkie tabele są dodane w schemacie "trip".

1. Stwórz aplikację typu REST Api.
2. Przygotuj końcówkę zwracającą dane z pomocą EntityFramework Core zgodnie z poniższymi informacjami:
  1. Końcówkę odpowiadającą na żądania HTTP GET wysyłane na adres `/api/trips`
  2. Końcówka powinna zwrócić listę podróży w kolejności posortowanej malejącą po dacie rozpoczęcia wycieczki.

### 3. Poniżej przedstawiony przykładowy format zwróconych danych.

```
{
  "Name": "ABC",
  "Description": "Lorem ipsum...",
  "DateFrom": "",
  "DateTo": "",
  "MaxPeople": 20,
  "Countries": [
    {
      "Name": "Poland"
    },
    {
      "Name": "Germany"
    }
  ],
  "Clients": [
    {
      "FirstName": "John",
      "LastName": "Smith"
    },
    {
      "FirstName": "Jake",
      "LastName": "Doe"
    }
  ]
}
```

### 3. Przygotuj końcówkę pozwalającą na usunięcie danych klienta.

1. Końcówka przyjmująca dane wysłane na adres HTTP DELETE na adres `/api/clients/{idClient}`
2. Końcówka powinna najpierw sprawdzić czy klient nie posiada żadnych przypisanych wycieczek. Jeśli klient posiada co najmniej jedną przypisaną wycieczkę – zwracamy błąd i usunięcie nie dochodzi do skutku.

### 4. Przygotuj końcówkę pozwalającą na przypisanie klienta do wycieczki.

1. Końcówka powinna przyjmować żądania http POST wysłane na adres `/api/trips/{idTrip}/clients`
2. Parametry przesłane w ciele żądania powinna wyglądać następująco (format danych może być inny):

```
{
  "FirstName": "John",
  "LastName": "Doe",
  "Email": "doe@wp.pl",
  "Telephone": "543-323-542",
  "Pesel": "91040294554",
  "IdTrip": 10,
  "TripName": "Rome",
  "PaymentDate": "4/20/2021"
}
```

### 5. Po przyjęciu danych sprawdzamy:

1. Czy klient o danym numerze PESEL istnieje. Jeśli nie, dodajemy go do bazy danych.
2. Czy klient nie jest już zapisaną na wspomnianą wycieczkę – w takim

wypadku zwracamy błąd.

3. Czy wycieczka istnieje – jeśli nie – zwracamy błąd.

6. „PaymentDate” może posiadać wartość null, dla tych klientów, którzy jeszcze nie zapłacili za podróż. Ponadto kolumna „RegisteredAt” w tabeli Client\_Trip powinna być równa aktualnemu czasowi przetworzenia żądania.
7. Pamiętaj o poprawnych nazwach zmiennych/metod/klas.
8. Wykorzystaj dodatkowe modele dla danych zwracanych i przyjmowanych przez końcówki – DTO (ang. Data Transfer Object).