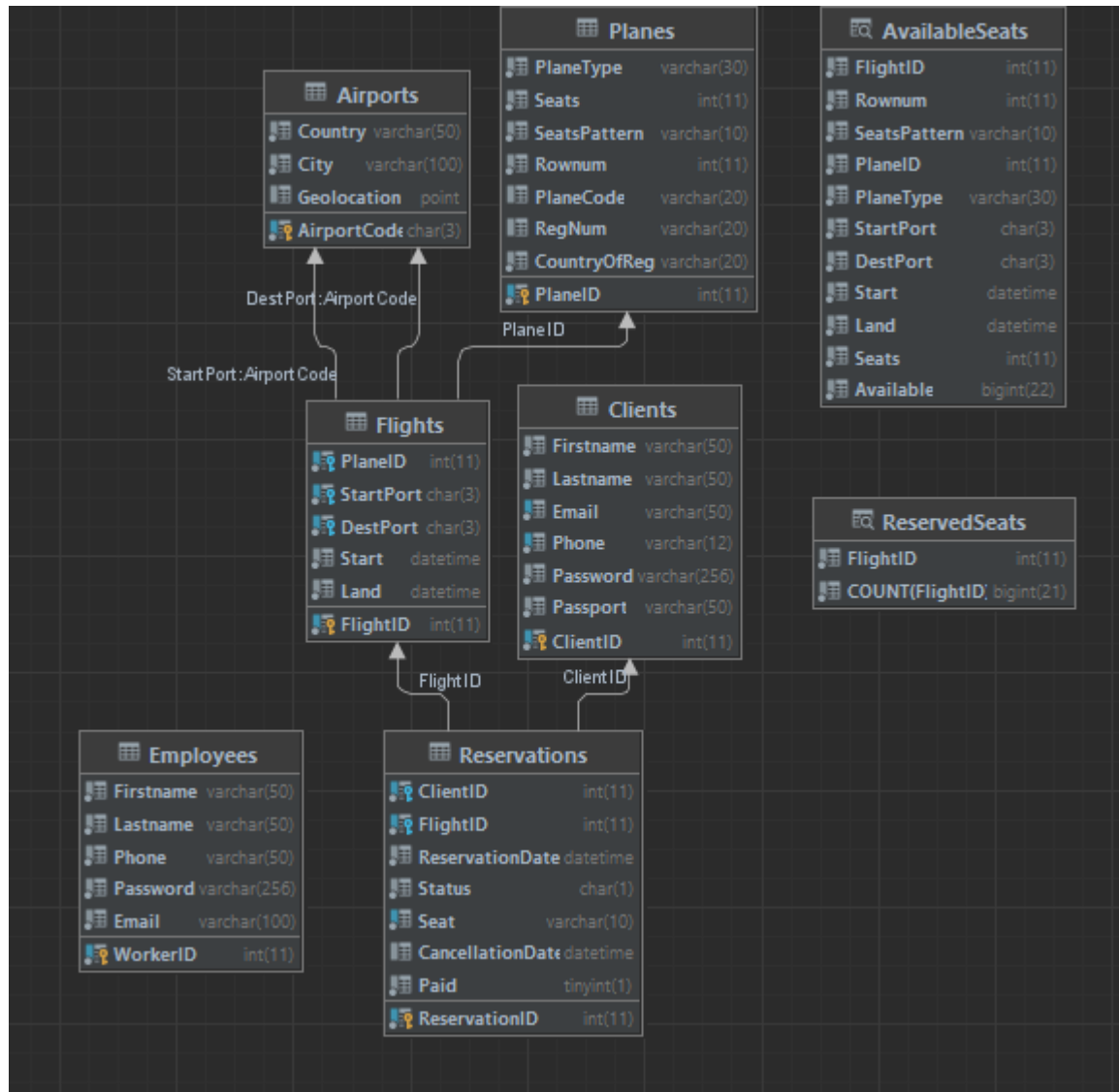


System do rezerwacji lotów - Szymon Woźniak, Mikołaj Pajor

Baza danych:

1. Schemat:



2. Tabele:

- Tabela Airports zawiera dane o lotniskach: 3-literowy kod lotniska, miasto oraz państwo, w którym się znajduje, oraz dane geolokalizacji, żeby znaleźć punkt na mapie.

```
create table Airports
(
    AirportCode char(3) not null
        primary key,
    Country varchar(50) not null,
    City varchar(100) not null,
    Geolocation point not null
);
```

- Tabela Flights zawiera dane dotyczące lotów: ID lotu, ID samolotu, który obsługuje lot, kody lotnisk startu i końca lotu oraz godziny startu i lądowania

```
create table Flights
(
    FlightID int auto_increment
        primary key,
    PlaneID int not null,
    StartPort char(3) not null,
    DestPort char(3) not null,
    Start datetime not null,
    Land datetime not null,
    constraint Flights_ibfk_2
        foreign key (StartPort) references Airports
(AirportCode),
    constraint Flights_ibfk_3
        foreign key (DestPort) references Airports
(AirportCode),
    constraint Flights_ibfk_4
        foreign key (PlaneID) references Planes (PlaneID)
);
```

- Tabela Planes zawiera szczegóły dotyczące samolotów, które obsługują lotów takie jak: ID samolotu, typ samolotu oraz jego numer, liczba miejsc oraz rozkład miejsc na pokładzie, liczba rzędów miejsc, numer rejestracyjny samolotu oraz kraj rejestracji

```
create table Planes
(
    PlaneID      int auto_increment
                primary key,
    PlaneType    varchar(30) not null,
    Seats        int         not null,
    SeatsPattern varchar(10) not null,
    Rownum       int         not null,
    PlaneCode    varchar(20) null,
    RegNum       varchar(20) null,
    CountryOfReg varchar(20) not null
);
```

- Tabela Clients przechowuje dane klientów, korzystających z naszej aplikacji: ID klienta, jego imię i nazwisko, numer paszportu, numer telefonu oraz email i hasło służące do logowania.

```
create table Clients
(
    ClientID int auto_increment
            primary key,
    Firstname varchar(50) not null,
    Lastname  varchar(50) not null,
    Email     varchar(50) not null,
    Phone     varchar(12) not null,
    Password  varchar(256) not null,
    Passport  varchar(50) not null,
    constraint Email
        unique (Email),
    constraint Phone
        unique (Phone)
);
```

- Tabela Reservations przechowuje dane dotyczące rezerwacji dokonanych w naszym systemie czyli: ID rezerwacji, ID lotu na który jest rezerwacja, ID klienta, który dokonał rezerwacji, datę rezerwacji, status rezerwacji (czy jest aktywna - 'A', czy jest anulowana - 'C'), ewentualną datę anulowania, status płatności oraz numer zarezerwowanego miejsca.

```
create table Reservations
(
    ReservationID    int auto_increment
                    primary key,
    ClientID         int          not null,
    FlightID         int          not null,
    ReservationDate  datetime     not null,
    Status           char         not null,
    Seat            varchar(10)   not null,
    CancellationDate datetime     null,
    Paid            tinyint(1)    not null,
    constraint only_one_seat
                    unique (FlightID, Seat),
    constraint Reservations_ibfk_1
                    foreign key (ClientID) references Clients
(ClientID),
    constraint Reservations_ibfk_3
                    foreign key (FlightID) references Flights
(FlightID)
                    on delete cascade
);
```

- Tabela Employees przechowuje informacje dotyczące pracowników naszego serwisu rezerwacji miejsc, mają oni dostęp do konta panelu administratora i mogą dodawać loty.

```
create table Employees
(
    WorkerID int auto_increment
            primary key,
    Firstname varchar(50) not null,
    Lastname  varchar(50) not null,
    Phone     varchar(50) not null,
    Password  varchar(256) not null,
    Email     varchar(100) not null
);
```

3. Widoki:

- Widok ReservedSeats zawiera informacje dotyczące liczby zarezerwowanych miejsc na dany lot:

```
create definer = swoznial@`%` view  
swoznial.ReservedSeats as  
select `swoznial`.`Reservations`.`FlightID`  
AS `FlightID`,  
      count(`swoznial`.`Reservations`.`FlightID`) AS  
`COUNT(FlightID)`  
from `swoznial`.`Reservations`  
where `swoznial`.`Reservations`.`Status` = 'A'  
group by `swoznial`.`Reservations`.`FlightID`;
```

	FlightID	COUNT(FlightID)
1	14	5
2	15	14
3	19	3

- Widok AvailableSeats jest zbiorem wszystkich danych, które były nam potrzebne do generowania zawartości strony. Jest połączeniem tabel Flights, Planes oraz widoku ReservedSeats.

```
create definer = swoznial@`%` view
swoznial.AvailableSeats as
select `F`.`FlightID` AS
`FlightID`,
      `P`.`Rownum` AS
`Rownum`,
      `P`.`SeatsPattern` AS
`SeatsPattern`,
      `F`.`PlaneID` AS
`PlaneID`,
      `P`.`PlaneType` AS
`PlaneType`,
      `F`.`StartPort` AS
`StartPort`,
      `F`.`DestPort` AS
`DestPort`,
      `F`.`Start` AS
`Start`,
      `F`.`Land` AS
`Land`,
      `P`.`Seats` AS
`Seats`,
      `P`.`Seats` - ifnull(`R`.`COUNT(FlightID)`, 0) AS
`Available`
from ((`swoznial`.`Flights` `F` join `swoznial`.`Planes`
`P`
      on (`F`.`PlaneID` = `P`.`PlaneID`)) left join
`swoznial`.`ReservedSeats` `R`
      on (`F`.`FlightID` = `R`.`FlightID`));
```

Powstał on po to, aby nie musieć pisać po stronie backend wielu długich kwerend, tylko jednym zapytaniem dostarczyć nam pełnych informacji o lotach.

Lista lotów - z perspektywy PRACOWNIKA

Strona Główna

Wyszukaj lot

Dodaj lot

Wyloguj się!

Lotnisko odlotu

dowolne

Lotnisko przylotu

dowolne

Numer lotu	Odlot	Przylot	Samolot	Dostępne Miejsca	Rezerwacja	Dla pracownika
24	JFK 15.06.2023, 23:36:00	IAH 30.06.2023, 23:36:00	Boeing 787-9 Dreamliner	198/198	Rezerwacja	Usuń lot
15	LHR 17.06.2023, 08:24:00	MAD 17.06.2023, 11:00:00	Airbus A330-243	269/280	Rezerwacja	Usuń lot
21	CDG 17.06.2023, 23:35:00	HND 30.06.2023, 23:35:00	Boeing 777-258(ER)	174/174	Rezerwacja	Usuń lot
16	MAD 20.06.2023, 12:12:00	PHX 21.06.2023, 01:12:00	Boeing 737-866	240/240	Rezerwacja	Usuń lot
19	CGK 5.07.2023, 12:00:00	HND 6.07.2023, 07:14:00	Airbus A330-243	280/280	Rezerwacja	Usuń lot
14	PVG 22.07.2023, 18:12:00	JFK 22.07.2023, 23:57:00	Boeing 787-9 Dreamliner	198/198	Rezerwacja	Usuń lot

Jak widać, na górnym pasku pojawia się dodatkowa możliwość - “Dodaj lot”

Strona Główna

Wyszukaj lot

Dodaj lot

Wyloguj się!

Lotnisko odlotu

wybierz lotnisko

Lotnisko przylotu

wybierz lotnisko

Samolot

wybierz samolot

Data/godzina odlotu

dd.mm.yyyy - :--

Data/godzina przylotu

dd.mm.yyyy - :--

Zaloguj się!

Tutaj możemy wybrać lotnisko (zarówno przylotu, jak i odlotu) z listy dostępnych lotnisk wprowadzonych do bazy danych do tabeli **Airports**

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
<input type="checkbox"/>	1	AirportCode	char(3)	utf8mb4_general_ci	Nie	Brak			Zmień Usun Więcej
<input type="checkbox"/>	2	Country	varchar(50)	utf8mb4_general_ci	Nie	Brak			Zmień Usun Więcej
<input type="checkbox"/>	3	City	varchar(100)	utf8mb4_general_ci	Nie	Brak			Zmień Usun Więcej
<input type="checkbox"/>	4	Geolocation	point		Tak	NULL			Zmień Usun Więcej

Lotnisko odlotu

wybierz lotnisko

wybierz lotnisko

AMS

ATL

BKK

CAN

CDG

CGK

DEN

DFW

DXB

Możemy także wybrać samolot w analogiczny sposób.

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
<input type="checkbox"/> 1	PlaneID	int(11)			Nie	Brak		AUTO_INCREMENT	Zmień Usuń Więcej
<input type="checkbox"/> 2	PlaneType	varchar(30)	utf8mb4_general_ci		Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/> 3	Seats	int(11)			Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/> 4	SeatsPattern	varchar(10)	utf8mb4_general_ci		Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/> 5	Rownum	int(11)			Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/> 6	PlaneCode	varchar(20)	utf8mb4_general_ci		Tak	NULL			Zmień Usuń Więcej
<input type="checkbox"/> 7	RegNum	varchar(20)	utf8mb4_general_ci		Tak	NULL			Zmień Usuń Więcej
<input type="checkbox"/> 8	CountryOfReg	varchar(20)	utf8mb4_general_ci		Nie	Brak			Zmień Usuń Więcej

Samolot

wybierz samolot

wybierz samolot
Boeing 737-866 (240)
Airbus A330-243 (280)
Boeing 777-258(ER) (174)
Airbus A319-132 (138)
Boeing 787-9 Dreamliner (198)
Airbus A380-842 (240)

Lista lotów - z perspektywy KLIENTA

[Strona Główna](#)
[Wyszukaj lot](#)
[Wyloguj się!](#)

Lotnisko odlotu Lotnisko przylotu

Numer lotu	Odlot	Przylot	Samolot	Dostępne Miejsca	Rezerwacja
24	JFK 15.06.2023, 23:36:00	IAH 30.06.2023, 23:36:00	Boeing 787-9 Dreamliner	198 / 198	Rezerwacja
15	LHR 17.06.2023, 08:24:00	MAD 17.06.2023, 11:00:00	Airbus A330-243	269 / 280	Rezerwacja
21	CDG 17.06.2023, 23:35:00	HND 30.06.2023, 23:35:00	Boeing 777-258(ER)	174 / 174	Rezerwacja
16	MAD 20.06.2023, 12:12:00	PHX 21.06.2023, 01:12:00	Boeing 737-866	240 / 240	Rezerwacja
19	CGK 5.07.2023, 12:00:00	HND 6.07.2023, 07:14:00	Airbus A330-243	280 / 280	Rezerwacja
14	PVG 22.07.2023, 18:12:00	JFK 22.07.2023, 23:57:00	Boeing 787-9 Dreamliner	198 / 198	Rezerwacja

Oczywiście wybór "Lotniska odlotu" oraz "Lotniska przylotu" realizujemy analogicznie do tego, jak zostało to zrealizowane w dodawaniu lotu.

Lotnisko odlotu Lotnisko przylotu

Numer lotu	Odlot	Przylot	Samolot	Dostępne Miejsca	Rezerwacja
19	CGK 5.07.2023, 12:00:00	HND 6.07.2023, 07:14:00	Airbus A330-243	272 / 280	Rezerwacja

Jest to tak naprawdę filtrowanie - zwracamy przeloty zgodnie z wskazaniami użytkownika.

Ten screenshot pokazuje po co jest nam potrzebne pole typu "point" - a mianowicie każde lotnisko ma przypisaną swoją geolokalizację. Koło wyboru lotniska odlotu / lotniska przylotu, mamy widoczne ikonki mapy, których kliknięcie powoduje wybór lotniska.

Strona Główna

Wyszukaj lot

Wyloguj się!

Lotnisko odlotu

dowolne

Lotnisko przylotu

dowolne

Mapa

Satelita

Numer lotu	Odlot	Przylot	Samolot	Dostępne Miejsca	Rezerwacja
24	JFK 15.06.2023, 23:36:00	IAH 30.06.2023, 23:36:00	Boeing 787-9 Dreamliner	198/198	Rezerwacja
15	LHR 17.06.2023, 08:24:00	MAD 17.06.2023, 11:00:00	Airbus A330-243	269/280	Rezerwacja
21	CDG 17.06.2023, 23:35:00	HND 30.06.2023, 23:35:00	Boeing 777-258(ER)	174/174	Rezerwacja
16	MAD 20.06.2023, 12:12:00	PHX 21.06.2023, 01:12:00	Boeing 737-866	240/240	Rezerwacja
19	CGK 5.07.2023, 12:00:00	HND 6.07.2023, 07:14:00	Airbus A330-243	272/280	Rezerwacja
14	PVG 22.07.2023, 18:12:00	JFK 22.07.2023, 23:57:00	Boeing 787-9 Dreamliner	198/198	Rezerwacja

Do pobrania tych danych zgodnie z filtrami korzystamy z widoku **AvailableSeats**, który zostanie omówiony w stosownym miejscu. Korzystamy z następującej kwerendy:

SELECT * FROM AvailableSeats WHERE Start > CURRENT_TIMESTAMP ZMIENNA ORDER BY Start ASC

Gdzie zmienna przyjmuje określoną wartość w zależności od wybranego filtra. Tutaj fragment kodu z backendu, który jest za to odpowiedzialny.

```
let sqlWhere = "";
if(req.params.from != "dowolne") {
    sqlWhere = ` AND StartPort = '${req.params.from}' `
}
if(req.params.to != "dowolne") {
    sqlWhere += ` AND DestPort = '${req.params.to}' `
}
```

Omówmy teraz działanie kont w aplikacji. Mamy tutaj dwie możliwości - konto klienta, oraz konto pracownika. Poniżej struktury obydwu tabeli w bazie danych.

Dla Klientów (Clients):

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
<input type="checkbox"/>	1	ClientID	int(11)		Nie	Brak		AUTO_INCREMENT	Zmień Usuń Więcej
<input type="checkbox"/>	2	Firstname	varchar(50)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	3	Lastname	varchar(50)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	4	Email	varchar(50)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	5	Phone	varchar(12)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	6	Password	varchar(256)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	7	Passport	varchar(50)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej

Dla Pracowników (Employees):

#	Nazwa	Typ	Metoda porównywania napisów	Atrybuty	Null	Ustawienia domyślne	Komentarze	Dodatkowo	Działanie
<input type="checkbox"/>	1	WorkerID	int(11)		Nie	Brak		AUTO_INCREMENT	Zmień Usuń Więcej
<input type="checkbox"/>	2	Firstname	varchar(50)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	3	Lastname	varchar(50)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	4	Phone	varchar(50)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	5	Password	varchar(256)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej
<input type="checkbox"/>	6	Email	varchar(100)	utf8mb4_general_ci	Nie	Brak			Zmień Usuń Więcej

Przy logowaniu aplikacja dopuszcza możliwość wyboru, czy logujemy się na konto pracownika, czy klienta. Oczywiście sama w sobie weryfikacja hasła następuje poprzez weryfikację **JWT Tokena**. Tutaj przykład takiego użytkownika na dowód, że hasła nie są przechowywane jako plain text, co znacznie podnosi bezpieczeństwo.

	ClientID	Firstname	Lastname	Email	Phone	Password	Passport
<input type="checkbox"/> Edytuj Kopiuji Usuń	226	test	test	test@test.test	test	\$2b\$10\$mVv4GLIYKZg8sJf03enareGC4hUwEM34xSsKk/AVDbB...	test

A tutaj formatka do logowania w aplikacji

Strona Główna

Logowanie

Rejestracja

Typ konta

Klient

Email

test

Hasło

....

Zaloguj sie!

Rezerwacja miejsc:

Po zalogowaniu się na konto użytkownika i wybraniu interesującego nas lotu mamy taki widok:

Strona Główna

Wyszukaj lot

Wyloguj się!

Numer lotu	Odlot	Przylot	Samolot	Dostępne Miejsca
16	MAD	PHX	Boeing 737-866	240/240
	20.06.2023, 12:12:00	21.06.2023, 01:12:00		

Rezerwacja miejsc

1	13	25	37	49	61	73	85	97	109	121	133	145	157	169	181	193	205	217	229
2	14	26	38	50	62	74	86	98	110	122	134	146	158	170	182	194	206	218	230
3	15	27	39	51	63	75	87	99	111	123	135	147	159	171	183	195	207	219	231
4	16	28	40	52	64	76	88	100	112	124	136	148	160	172	184	196	208	220	232
5	17	29	41	53	65	77	89	101	113	125	137	149	161	173	185	197	209	221	233
6	18	30	42	54	66	78	90	102	114	126	138	150	162	174	186	198	210	222	234
7	19	31	43	55	67	79	91	103	115	127	139	151	163	175	187	199	211	223	235
8	20	32	44	56	68	80	92	104	116	128	140	152	164	176	188	200	212	224	236
9	21	33	45	57	69	81	93	105	117	129	141	153	165	177	189	201	213	225	237
10	22	34	46	58	70	82	94	106	118	130	142	154	166	178	190	202	214	226	238
11	23	35	47	59	71	83	95	107	119	131	143	155	167	179	191	203	215	227	239
12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228	240

Wybrane miejsca
Brak wybranych miejsc...

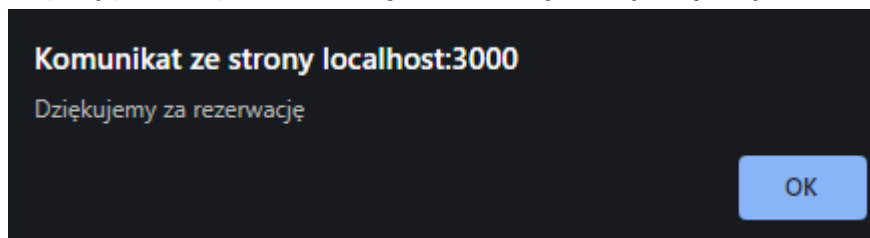
Kliknij, aby dokonać rezerwacji

Żeby zarezerwować miejsca zaznaczamy interesujące nas miejsca i naciskamy przycisk.

Wybrane miejsca
238 239 240

Kliknij, aby dokonać rezerwacji

W przypadku prawidłowej rezerwacji otrzymujemy komunikat:



```
for (let seat of req.body.seats){
  let query = `INSERT INTO Reservations(ClientID,FlightID,Paid,ReservationDate,Seat,Status)
              VALUE (${req.body.ClientID},${req.body.FlightID},1,NOW(),${seat},'A');`
  let gresult= await reserveSeat(query);
  if(!gresult) throw new Error( message: "Something's wrong");
}
```

Ponieważ w tablicy Reservations kombinacja miejsce-lot jest unikalna, nie zarezerwujemy tego miejsca dwa razy. Dodatkowo jako że przyjmujemy taktykę wszystko albo nic, cała rezerwacja objęta jest transakcją:

```
await sql.query('SET TRANSACTION ISOLATION LEVEL SERIALIZABLE');

try{
  await sql.beginTransaction();
  for (let seat of req.body.seats){
    let query = `INSERT INTO Reservations(ClientID,FlightID,Paid,ReservationDate,Seat,Status)
                VALUE (${req.body.ClientID},${req.body.FlightID},1,NOW(),${seat},'A');`
    let gresult= await reserveSeat(query);
    if(!gresult) throw new Error( message: "Something's wrong");
  }
  await sql.commit();
  res.send( body: {status:"OK"})
}catch (err){
  console.error(`Error occurred while creating order: ${err.message}`, err);
  sql.rollback();
  res.send( body: {error:err.message});
}
```

Wybraliśmy poziom izolacji SERIALIZABLE żeby maksymalnie uniknąć wszelkich błędów (chodzi jednak o pieniądze klientów i miejsca w samolocie). W przypadku zwrócenia błędu przez reserveSeat() dokonujemy rollbacku transakcji i wszelkie rezerwacje są wycofywane. W przypadku dodania wszystkich rezerwacji transakcja jest commitowana a miejsca zarezerwowane. Jeśli transakcja zostanie wycofana, nasz klient otrzymuje komunikat:

