

Wrocław 23.05.2018

Jan Venulet 235542

Artur Frąćala 235599

Kamil Kucharski 235014 (lider)

Grzegorz Otto 234941

Grupa 23

Temat projektu: System rezerwacji lotów i
obsługi lotniska

Spis treści

1. Spis narzędzi użytych przy tworzeniu projektu	3
2. Określenie zakresu i kontekstu projektu.	3
3. Identyfikacja wymagań funkcjonalnych	7
4. Model logiczny bazy danych.....	8
a) Legenda diagramu	8
b) Przykładowe połączenia wraz z legendą połączeń diagramu.....	9
c) Diagram ERD	10
5. Skrypt SQL tworzący strukturę bazy danych i wypełniający ją danymi	11
a) Utworzenie wszystkich tabel naszej bazy:.....	11
b) Utworzenie sekwencji do każdej z tabel:	15
c) Utworzenie widoku:	17
d) Operacje wstawiania danych do wszystkich tabel w odpowiedniej kolejności:	17
e) Utworzenie tabeli logów	21
e) Utworzenie triggera do wywoływania logów.....	22

1. Spis narzędzi użytych przy tworzeniu projektu

- SQL Developer
- NetBeans IDE 8.2
- draw.io

2. Określenie zakresu i kontekstu projektu.

Nasz system będzie miał na celu obsługę jednego lotniska, które będzie również świadczyło usługi zakupu biletów oraz rezerwacji bagażów na konkretny lot jednej z linii odlatujących z naszego lotniska. Będzie to system pośredniczący między pasażerami, liniami lotniczymi oraz lotniskiem. Pasażer w domyśle nie będzie miał możliwości zakupu bilet na inny lot niż ten odlatujący z naszego lotniska, ponieważ założenie bazuje na tym iż każde inne lotnisko ma swój system rezerwacji biletów i bagażów, a poszczególne linie lotnicze potrzebują uzyskać dostęp do każdego z obsługiwanych lotnisk i tam zarezerwować dane lotu z możliwej na lotnisku floty oraz dostępnej daty i godziny.

Role dostępne w naszym systemie można podzielić na czterech zalogowanych użytkowników oraz tryb ogólnodostępny dla każdego z nich.

a). Dla trybu ogólnodostępnego, w pewnym sensie będzie to strona główna programu gdzie będzie wyświetlana tablica najbliższych odlotów i przylotów. Będą to połączenia które pracownik linii doda do tabeli odlot lub przylot, a pracownik lotniska zaakceptuje to połączenie. Jeżeli użytkownik będzie chciał sprawdzić czy dany lot jest obsługiwany w jakieś konkretne miejsce, lub kiedy może się spodziewać przylotu, bądź też jakie loty będą obsługiwane w dniu jego urlopu to po przejściu do funkcjonalności wyszukaj lot, będzie mógł wybrać filtry wyszukiwania, a poniżej przedstawi mu się tabela odlotów bądź też przylotów, które go interesują.

Z okna startowego będzie dostępna możliwość przejścia to trybu dostępu dla konkretnego aktora. Dla klientów którzy mają na celu zakupić bilet na lot, będzie

to okno zaloguj jako pasażer. Tutaj w zależności od tego, czy klient posiada już konto, lub nie, będzie mógł przejść przez proces rejestracji lub też bezpośrednio przejść do logowania, w wypadku kiedy posiada już konto w serwisie naszego lotniska.

Podczas procesu rejestracji zostanie dodany do tabeli users rekord użytkownika o loginie i hasle wybranym przez użytkownika, o trybie dostępu tylko dla pasażerów bez plenipotencji dostępnych pracownikom lotniska. Oczywiście system będzie sprawdzał czy zostały spełnione kryteria, którymi będą: niepowtarzalność loginów oraz ogórne wymagania dotyczące hasła, np. wymagana długość. Podczas tego procesu zostanie dodany do tabeli pasażerowie rekord z danymi osobowymi podanymi przez pasażera . Będą to imię, nazwisko, data urodzenia, numer paszportu i dane kontaktowe, takie jak numer telefonu i e-mail.

Po zakończeniu procesu rejestracji użytkownik wróci do poprzedniego okna, z którego już będzie mógł przejść do pola logowania gdzie po prawidłowym podaniu uprzednio wprowadzonych danych zostanie przekierowany do dostępnego dla siebie trybu pasażera.

Proces logowania jest identyczny dla pracowników lotniska oraz linii lotniczych.

Różnica w wyniku logowania polega na tym, iż w zależności od podanego loginu i hasła, ukaże się inny wachlarz możliwości, odpowiadający trybowi dostępu.

Pracownicy lotniska i linii nie mają możliwości dodania konta przez samego siebie. Dane do konta udostępniane są mu poprzez osobę z najwyższym trybem dostępu jakim jest pracownik działu logistyki lotniska.

b). Pracownik działu logistyki lotniska będzie w zamyśle projektowym najwyżej położonym w hierarchii użytkownikiem systemu lotniska. On będzie miał możliwość utworzenia kont dla innych pracowników lotniska, czy linii lotniczych korzystających z lotniska. Głównym zadaniem pracownika logistyki lotniska będzie zatwierdzanie rezerwacji lotów złożonych przez pracowników linii lotniczych, w zależności od możliwości grafika oraz dostępnych pracowników i przepustowości bramek. Przy procesie zatwierdzania kluczowym zadaniem będzie potrzeba przypisania pracowników obsługi lotniska do lotu oraz przypisania bramki która będzie obsługiwała lot. Przy procesie przypisywania pracowników zostanie wstawiony rekord do tabeli przydziałów, gdzie pracownik i lot zostaną zawarte, tak aby jeden pracownik był w stanie obsługiwać parę

lotów jednego dnia, a lot będzie mógł posiadać dowolną ilość pracowników go obsługujących.

Edycja bazy pracowników również będzie jedną z funkcjonalności pracownika działu obsługi lotniska. Może on w zależności od potrzeby uaktualnić dane obecnych pracowników, dodać nowego pracownika do bazy, lub też go usunąć. Będzie on miał również dostęp do wglądu na siatkę połączeń lotniska oraz w zależności od potrzeby, zmiany statusu lotu (opóźniony o [ilość czasu], odwołany, odleciał).

c). Pracownik linii lotniczych loguje się na podstawie danych udostępnionych mu poprzez pracownika działu logistyki lotniska. Jego funkcjonalność i możliwości są mniejsze niż pracowników tego działu. Ten konkretny rodzaj konta będzie miał możliwość dodania przylotów i odlotów tylko obsługiwanych przez to lotnisko. Korzysta on z systemu na zasadzie utworzenia lotu i złożenia zapytania do lotniska, czy jest ono w stanie obsłużyć lot o podanej dacie i godzinie. Przy tworzeniu połączenia pracownik linii będzie musiał podać samolot ,z dostępnej przez jego linię floty, który będzie miał za zadanie obsłużyć dany rejs oraz przypisać mu desygnowanych pracowników. Podczas tego procesu będzie również trzeba określić cenę jaką sobie żąda przewoźnik za jeden bilet na ten rejs. Zmiana ceny nie będzie możliwa. W założeniu projektu jest dostępna tylko jedna kategoria cenowa na każdy przelot.

Bazę pracowników konkretnej linii, posiadanej floty i tabelę przydziałów personelu do lotów również będzie można edytować z trybu dostępu pracownika linii w zależności od zaistniałych, nieprzewidzianych potrzeb.

Ponieważ klienci będą również mieli możliwość złożenia reklamacji, której wachlarz będzie ogólnie ustalony, pracownik linii lotniczej będzie miał również dostęp do wglądu na tabelę zażaleń wraz z ich rodzajem, i przez jakiego klienta została złożona. W zależności od zasadności reklamacji będzie możliwość jej przyjęcia bądź też odrzucenia, co pasażer ją składający będzie mógł sprawdzić za pomocą swojego konta. Przyjętym założeniem przy procesie reklamacji w wypadku jej akceptowania jest to iż inny organ/dział finansowy konkretnych linii zajmuje się zwrotem kosztów czy też oszacowaniem i wypłaceniem odszkodowania. Nasz system nie służy w tym celu więc jego funkcjonalność nie zawiera dalszych kroków obsługi tychże reklamacji.

Pracownik linii będzie również mógł zmienić status już zaakceptowanych przez

lotnisko lotów.

Będzie również mógł zmienić akceptowane przez jego linię restrykcje dotyczące bagażu. Będą one zawierać trzy wymiary wyrażane w centymetrach oraz wagę bagażu w wyrażoną w kilogramach.

Poza tymi funkcjonalnościami będzie miał również zmienić krotkę hasła w rekordzie swojego konta w bazie users (użytkownicy), za pomocą której będzie się logował.

d). Pracownik lotniska obsługujący bagaż loguje się na podstawie danych udostępnionych mu poprzez pracownika działu logistyki lotniska. Jego zakres obowiązków i możliwości oferowanych przez system będzie dotyczył sprawdzenia czy obecny pasażer przy punkcie nadania bagażu dokonał rezerwacji bagażu przy zakupie biletu, na podstawie jego danych, fizycznego sprawdzenia wymiarów bagażu oraz wprowadzenia ich do systemu. Jeżeli bagaż przekracza jeden z wymiarów lub też wagę wymaganą przez linię lotniczą, obsługujący ten konkretny lot na który jest nadawany bagaż, zostanie automatycznie wprowadzenie bitu nadbagażu, odpowiadającemu za identyfikowanie bagażu przekraczającego przyjęte przez linię lotniczą normy. Nie ma rozróżnienia na kategorie nadbagażu bazującej o ile dany limit został przekroczony.

e). Pasażer podróżujący z naszego lotniska po zalogowaniu się na konto które zostało przez niego uprzednio utworzone, ma możliwość wyszukania lotu odlatującego z naszego lotniska oraz zakupienia go. Każdy pasażer może kupić tylko jeden bilet na każdy lot. System nie blokuje zakupu paru biletów na wiele lotów tego samego dnia, ponieważ to w interesie klienta jest zadbanie o logiczny zakup biletów. Baza wyszukiwanych lotów będzie utworzona na podstawie lotów dodanych przez pracownika linii i następnie zatwierdzonych przez pracownika działu logistyki lotniska. Loty dodane przez linię, ale nie zatwierdzone nie będą widoczne dla pasażerów. Pasażer szuka lotów na podstawie ustalonych przez siebie kryteriów. Kiedy już pasażer zdecyduje się na konkretny lot o podanej cenie przejdzie do okna zakupu. Jeżeli lot nie będzie już miał miejsca, pasażer będzie mógł odszukać lot, ale nie będzie możliwe dla niego wejście w tryb zakupu. W oknie zakupu pasażer będzie mógł zdecydować się na jedno z dostępnych miejsc. Lista dostępnych miejsc będzie tworzona na

podstawie wielkości samolotu oraz listy biletów innych pasażerów którzy już zakupili bilet z konkretnym miejscem. Pasażer również w oknie zakupu deklaruje czy chce zabrać ze sobą bagaż nadany, który zostanie mu przypisany do biletu.

W założeniu ilość miejsca dostępna w luku bagażowym jest precyzyjnie ustalona i na jeden bilet może przypadać maksymalnie jeden bagaż.

Bagaż podręczny pasażera jest wliczony w cenę biletu i jego zakup jest obligatoryjny. Ponieważ lotnisko tylko fizycznie sprawdza zawartość bagażu podręcznego, a następnie linia obsługująca lot będzie wprowadzać pasażerów na pokład samolotu wraz z ich bagażem, to w zakresie ich obowiązków jest sprawdzenie czy bagaż podręczny nie przekracza ich uniwersalnych norm obowiązujących dla całej linii. Lotnisko dostarcza tylko usługi dotyczące bagażu nadawanego. W domyśle linia sama odgórnie będzie musiała zająć się dowolnie od ich interpretacji nadbagażem podręcznym.

Pasażer również może sprawdzić na jakie loty zakupił bilet oraz czy wykupił na nie bagaż nadawany na lotnisku.

Poza zakupem biletu i sprawdzeniem stanu biletów, klient może również złożyć reklamacje/ zażalenie obsługiwaną przez linię lotniczą. Tylko klienci którzy mają bilet na lot obsługiwany przez nasze lotnisko będą mogli złożyć reklamację. Po złożeniu reklamacji otrzymają oni indywidualny kod, na podstawie którego będą mogli sprawdzić status złożonej przez siebie reklamacji, czy została ona zaakceptowana, odrzucona bądź cały czas jest w stanie oczekującym na decyzję linii.

3. Identyfikacja wymagań funkcjonalnych

Aktor 1 [pasażer]

Funkcje:

- wyszukanie lotu
- rezerwacja lotu wraz z miejscem i bagażem
- składanie reklamacji
- sprawdzenie statusu reklamacji

Aktor 2[pracownik linii lotniczej]

Funkcje:

- edycja floty
- rezerwacja połączeń lotniczych
- przypisywanie personelu do obsługi lotu
- edycja bazy pracowników linii
- akceptowanie lub odrzucanie reklamacji

Aktor 3 [pracownik działu logistycznego lotniska]

Funkcje:

- akceptowanie lub odrzucanie połączeń lotniczych
- przydział samolotów i pasażerów do odpowiednich bramek
- sprawdzenie siatki połączeń
- przydział personelu do obsługi lotniska
- edycja bazy pracowników lotniska

Aktor 4 [pracownik lotniska obsługujący bagaż]

Funkcje:

- zmiana statusu nadania bagażu
- nadanie wagi bagażom

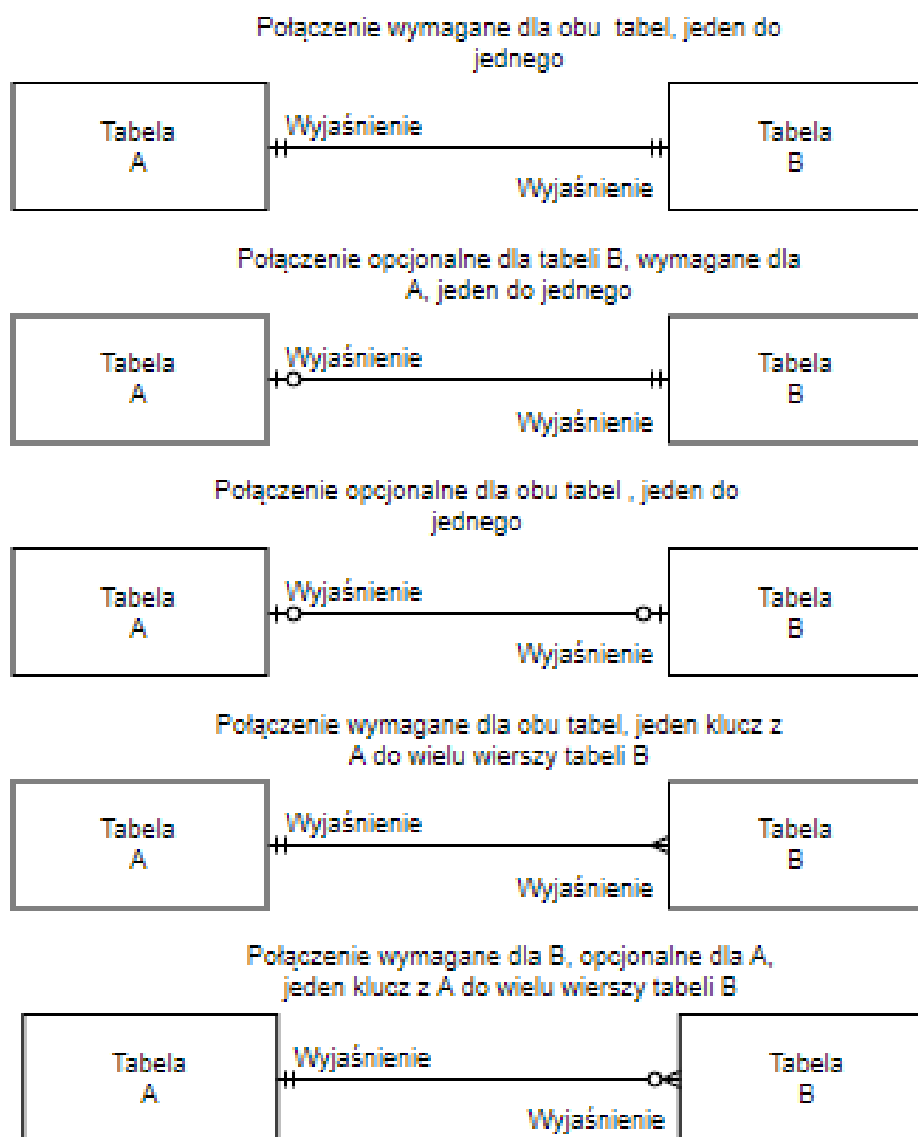
4. Model logiczny bazy danych

a) Legenda diagramu

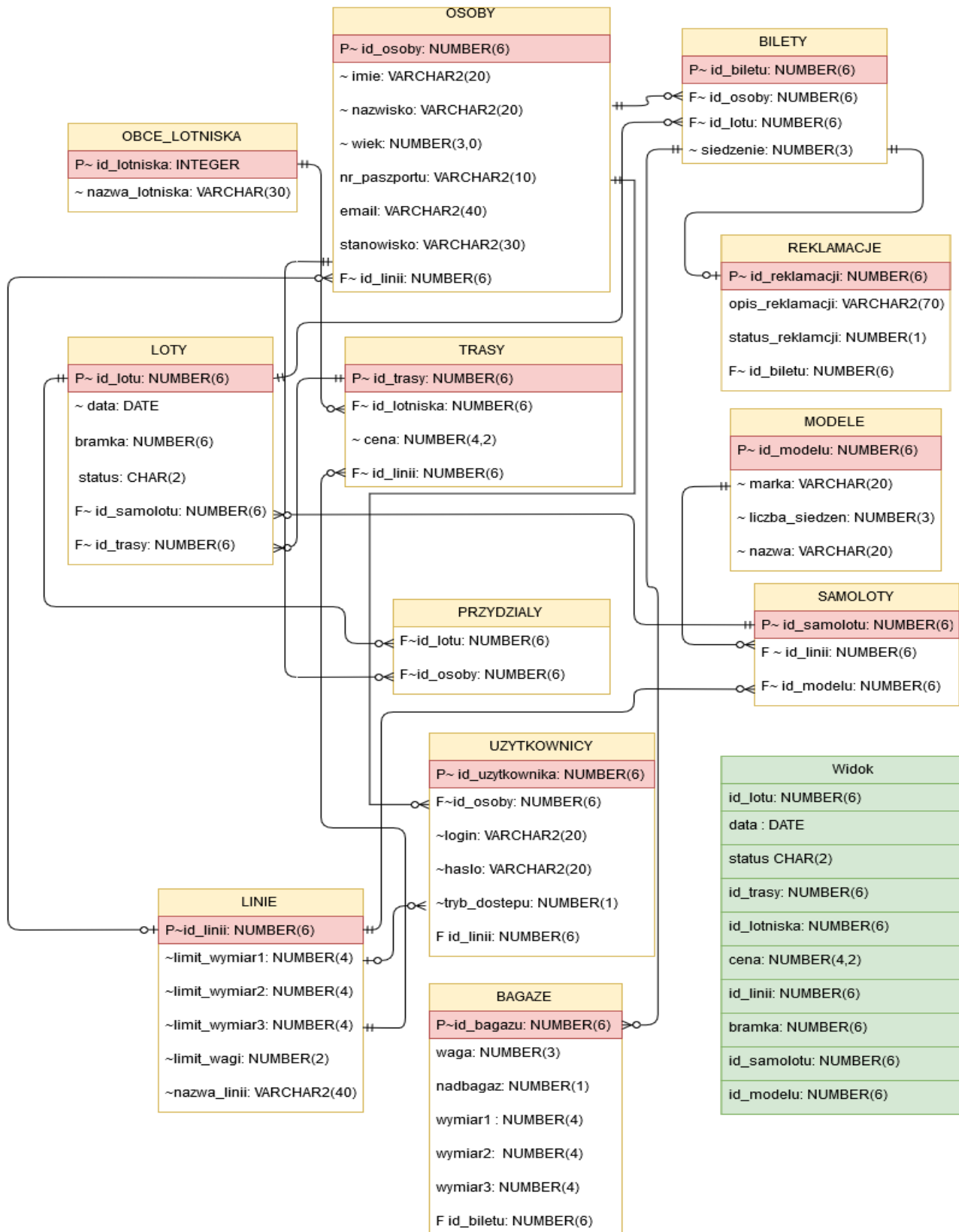
- Litera P (Priamary) oznacza, że dane pole w tabeli jest kluczem głównym
- Litera F (Foreign) oznacza, że dane pole w tabeli jest kluczem obcym
- Znak ' ~ ' oznacza, iż dane pole w tabeli jest NOT NULL

Do stworzenia diagramu ERD użyliśmy notacji Martina

b) Przykładowe połączenia wraz z legendą połączeń diagramu



c) *Diagram ERD*



5. Skrypt SQL tworzący strukturę bazy danych i wypełniający ją danymi

a) Utworzenie wszystkich tabel naszej bazy:

```
CREATE TABLE MODELE
```

```
(  
    id_modelu NUMBER(6,0) PRIMARY KEY,  
    marka VARCHAR2(20) NOT NULL,  
    liczba_siedzen NUMBER(3) NOT NULL CHECK (liczba_siedzen > 0),  
    nazwa VARCHAR2(20) NOT NULL UNIQUE  
);
```

```
CREATE TABLE OBCE_LOTNISKA
```

```
(  
    id_lotniska NUMBER(6,0) PRIMARY KEY,  
    nazwa_lotniska VARCHAR(30) NOT NULL UNIQUE  
);
```

```
CREATE TABLE LINIE
```

```
(  
    id_linii NUMBER(6,0) PRIMARY KEY,  
    limit_wymiar1 NUMBER(4,0) NOT NULL CHECK (limit_wymiar1>0) ,  
    limit_wymiar2 NUMBER(4,0) NOT NULL CHECK (limit_wymiar2 > 0 ),  
    limit_wymiar3 NUMBER(4,0) NOT NULL CHECK (limit_wymiar3 > 0 ),
```

```
limit_wagi NUMBER(2,0) NOT NULL CHECK (limit_wagi > 0 ),  
nazwa_linii VARCHAR2(40) NOT NULL UNIQUE  
);
```

```
CREATE TABLE SAMOLOTY
```

```
(  
    id_samolotu NUMBER(6,0) PRIMARY KEY,  
    id_linii NUMBER(6,0) NOT NULL REFERENCES LINIE(id_linii),  
    id_modelu NUMBER(6,0) NOT NULL REFERENCES MODELE(id_modelu)  
);
```

```
CREATE TABLE OSOBY
```

```
(  
    id_osoby NUMBER(6,0) PRIMARY KEY,  
    imie VARCHAR(20) NOT NULL,  
    nazwisko VARCHAR(40) NOT NULL,  
    wiek NUMBER(3,0) NOT NULL CHECK (wiek > 0 AND wiek <150),  
    nr_paszportu VARCHAR2(10) UNIQUE,  
    email VARCHAR2(40) UNIQUE,  
    stanowisko VARCHAR2(30),  
    id_linii NUMBER(6,0) REFERENCES LINIE(id_linii)  
);
```

```
CREATE TABLE TRASY
```

```
(
  id_trasy NUMBER(6,0) PRIMARY KEY,
  cena NUMBER(4,0) CHECK (cena > 0),
  id_lotniska NUMBER(6,0) NOT NULL REFERENCES obce_lotniska(id_lotniska) ,
  id_linii NUMBER(6,0) REFERENCES linie(id_linii)
);
```

CREATE TABLE UZYTKOWNICY

```
(
  id_uzytkownika NUMBER(6,0) PRIMARY KEY,
  id_osoby NUMBER(6,0) NOT NULL,
  login VARCHAR2(20) NOT NULL UNIQUE,
  haslo VARCHAR2(20) NOT NULL,
  tryb_dostepu NUMBER(1) DEFAULT 0 NOT NULL,
  id_linii NUMBER(6) REFERENCES LINIE(id_linii)
);
```

CREATE TABLE LOTY

```
(
  id_lotu NUMBER(6,0) PRIMARY KEY,
  data DATE NOT NULL,
  bramka NUMBER(6,0),
  status CHAR(2),
  id_samolotu NUMBER(6,0) NOT NULL REFERENCES SAMOLOTY(id_samolotu),
  id_trasy NUMBER(6,0) NOT NULL REFERENCES TRASY(id_trasy)
);
```

CREATE TABLE PRZYDZIALY

```
(  
    id_lotu NUMBER(6) NOT NULL REFERENCES LOTY(id_lotu),  
    id_osoby NUMBER(6) NOT NULL REFERENCES OSOBY(id_osoby)  
);
```

CREATE TABLE BILETY

```
(  
    id_biletu NUMBER(6,0) PRIMARY KEY,  
    id_osoby NUMBER(6,0) NOT NULL REFERENCES OSOBY(id_osoby),  
    id_lotu NUMBER(6,0) NOT NULL REFERENCES LOTY(id_lotu),  
    siedzenie NUMBER(3) NOT NULL CHECK(siedzenie > 0)  
);
```

CREATE TABLE BAGAZE

```
(  
    id_bagazu NUMBER(6,0) PRIMARY KEY,  
    waga NUMBER(6,0) CHECK (waga > 0),  
    nadbagaz NUMBER(1,0) DEFAULT 0,  
    wymiar1 NUMBER(4,0) CHECK (wymiar1 > 0),  
    wymiar2 NUMBER(4,0) CHECK (wymiar2 > 0),  
    wymiar3 NUMBER(4,0) CHECK (wymiar3 > 0),  
    id_biletu NUMBER(6,0) REFERENCES BILETY(id_biletu)  
);
```

CREATE TABLE REKLAMACJE

```
(  
    id_reklamacji NUMBER(6,0),  
    opis_reklamacji VARCHAR2(70),  
    status_reklamacji NUMBER(1,0),  
    id_biletu NUMBER(6,0) NOT NULL REFERENCES BILETY (id_biletu)  
);
```

b) Utworzenie sekwencji do każdej z tabel:

```
CREATE SEQUENCE uzytkownicy_seq  
INCREMENT BY 1  
START WITH 1  
NOCACHE  
NOCYCLE;
```

```
CREATE SEQUENCE osoby_seq  
INCREMENT BY 1  
START WITH 1  
NOCACHE  
NOCYCLE;
```

```
CREATE SEQUENCE obce_lotniska_seq  
INCREMENT BY 1  
START WITH 1  
NOCACHE  
NOCYCLE;
```

```
CREATE SEQUENCE modele_seq
```

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE samoloty_seq

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE linie_seq

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE trasy_seq

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;

CREATE SEQUENCE reklamacje_seq

INCREMENT BY 1

START WITH 1

NOCACHE

NOCYCLE;


```
CREATE SEQUENCE bagaze_seq  
INCREMENT BY 1  
START WITH 1  
NOCACHE  
NOCYCLE;
```

```
CREATE SEQUENCE bilety_seq  
INCREMENT BY 1  
START WITH 1  
NOCACHE  
NOCYCLE;
```

```
CREATE SEQUENCE loty_seq  
INCREMENT BY 1  
START WITH 1  
NOCACHE  
NOCYCLE;
```

c) Utworzenie widoku:

```
CREATE VIEW WIDOK AS  
SELECT * FROM LOTY NATURAL JOIN TRASY;
```

d) Operacje wstawiania danych do wszystkich tabel w odpowiedniej kolejności:

```

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL,
'Santiago');

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL,
'Puebla');

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL,
'Lizbona');

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL,
'Alicante');

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL,
'Cancun');

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL, 'Lima');
insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL, 'Mexico
City');

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL, 'Vigo');
insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL,
'Sewilla');

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL,
'Bogota');

insert into OBCE_LOTNISKA (id_lotniska, nazwa_lotniska) VALUES (obce_lotniska_seq.NEXTVAL, 'Miami');


insert into LINIE (id_linii, limit_wymiar1, limit_wymiar2, limit_wymiar3, limit_wagi, nazwa_linii) VALUES
(LINIE_seq.NEXTVAL, 120, 50, 42, 20, 'Iberia');

insert into LINIE (id_linii, limit_wymiar1, limit_wymiar2, limit_wymiar3, limit_wagi, nazwa_linii) VALUES
(LINIE_seq.NEXTVAL, 80, 40, 38, 18, 'Peruvian Airlines');

insert into LINIE (id_linii, limit_wymiar1, limit_wymiar2, limit_wymiar3, limit_wagi, nazwa_linii) VALUES
(LINIE_seq.NEXTVAL, 105, 52, 50, 23, 'LATAM Airlines');

insert into LINIE (id_linii, limit_wymiar1, limit_wymiar2, limit_wymiar3, limit_wagi, nazwa_linii) VALUES
(LINIE_seq.NEXTVAL, 90, 60, 36, 20, 'AirFrance');

insert into LINIE (id_linii, limit_wymiar1, limit_wymiar2, limit_wymiar3, limit_wagi, nazwa_linii) VALUES
(LINIE_seq.NEXTVAL, 90, 60, 36, 20, 'AeroMexico');

```

```
INSERT INTO OSOBY VALUES(OSOBY_SEQ.NEXTVAL, 'Tomasz', 'Zieba', 25, 1234, 'tomasz@zieba.com', 3, NULL);
```

```
INSERT INTO OSOBY VALUES(OSOBY_SEQ.NEXTVAL, 'Andrzej', 'Krzak', 40, 1000, 'andrzej123@gmail.com', NULL, NULL);
```

```
INSERT INTO OSOBY VALUES(OSOBY_SEQ.NEXTVAL, 'Zbigniew', 'Zimoch', 23, 53141, 'zzimon@hotmail.com', 1, NULL);
```

```
INSERT INTO OSOBY VALUES(OSOBY_SEQ.NEXTVAL, 'Anna', 'Grzyb', 11, 4141, 'agrz@mit.usa', NULL, NULL);
```

```
INSERT INTO OSOBY VALUES(OSOBY_SEQ.NEXTVAL, 'Joachim', 'Lew', 42, 5315, 'achimek1976@gmail.com', NULL, NULL);
```

```
INSERT INTO UZYTKOWNICY VALUES(UZYTKOWNICY_SEQ.NEXTVAL, 4, 'pusia123', 'haslo123', 0, NULL);
```

```
INSERT INTO UZYTKOWNICY VALUES(UZYTKOWNICY_SEQ.NEXTVAL, 3, 'zzimon', 'zimon', 0, NULL);
```

```
INSERT INTO UZYTKOWNICY VALUES(UZYTKOWNICY_SEQ.NEXTVAL, 2, 'krzaczysko', 'ognisko', 0, NULL);
```

```
INSERT INTO UZYTKOWNICY VALUES(UZYTKOWNICY_SEQ.NEXTVAL, 5, 'achim1976', 'haslo', 0, NULL);
```

```
INSERT INTO UZYTKOWNICY VALUES(UZYTKOWNICY_SEQ.NEXTVAL, 1, 'tomzieb', 'l4tjklna', 0, NULL);
```

```
INSERT INTO MODELE VALUES (MODELE_SEQ.NEXTVAL, 'Tupolew' , 210, 'Tu-204' );
```

```
INSERT INTO MODELE VALUES (MODELE_SEQ.NEXTVAL, 'Airbus' , 150, 'A320' );
```

```
INSERT INTO MODELE VALUES (MODELE_SEQ.NEXTVAL, 'Boeing' , 104, '737' );
```

```
INSERT INTO MODELE VALUES (MODELE_SEQ.NEXTVAL, 'Airbus' , 560, 'A380' );
```

```
INSERT INTO MODELE VALUES (MODELE_SEQ.NEXTVAL, 'Boeing' , 250, '787 Dreamliner' );
```

```
INSERT INTO SAMOLOTY VALUES ( SAMOLOTY_SEQ.NEXTVAL, 2, 2);
```

```
INSERT INTO SAMOLOTY VALUES ( SAMOLOTY_SEQ.NEXTVAL, 2, 5);
```

```
INSERT INTO SAMOLOTY VALUES ( SAMOLOTY_SEQ.NEXTVAL, 5, 4);
```

```
INSERT INTO SAMOLOTY VALUES ( SAMOLOTY_SEQ.NEXTVAL, 1, 3);
```

```
INSERT INTO SAMOLOTY VALUES ( SAMOLOTY_SEQ.NEXTVAL, 1, 4);
```

```
INSERT INTO TRASY VALUES (TRASY_SEQ.NEXTVAL, 429 ,5, 1);
```

INSERT INTO TRASY VALUES (TRASY_SEQ.NEXTVAL, 429 ,6, 2);

INSERT INTO TRASY VALUES (TRASY_SEQ.NEXTVAL, 429 ,2, 1);

INSERT INTO TRASY VALUES (TRASY_SEQ.NEXTVAL, 429 ,7, 1);

INSERT INTO TRASY VALUES (TRASY_SEQ.NEXTVAL, 429 ,1, 3);

INSERT INTO LOTY VALUES (LOTY_SEQ.NEXTVAL, to_date('20180623 11:20','yyyymmdd hh24:mi'), 2, 'ap', 1, 1);

INSERT INTO LOTY VALUES (LOTY_SEQ.NEXTVAL, to_date('20180703 9:50','yyyymmdd hh24:mi'), 2, 'ap', 2, 3);

INSERT INTO LOTY VALUES (LOTY_SEQ.NEXTVAL, to_date('20180619 15:00','yyyymmdd hh24:mi'), 3, 'oo', 1, 1);

INSERT INTO LOTY VALUES (LOTY_SEQ.NEXTVAL, to_date('20180618 11:20','yyyymmdd hh24:mi'), 1, 'ao', 2, 2);

INSERT INTO LOTY VALUES (LOTY_SEQ.NEXTVAL, to_date('20180619 18:20','yyyymmdd hh24:mi'), 5, 'ao', 2, 4);

INSERT INTO BILETY VALUES(BILETY_SEQ.NEXTVAL, 2, 1, 22);

INSERT INTO BILETY VALUES(BILETY_SEQ.NEXTVAL, 2, 2, 24);

INSERT INTO BILETY VALUES(BILETY_SEQ.NEXTVAL, 1, 3, 27);

INSERT INTO BILETY VALUES(BILETY_SEQ.NEXTVAL, 4, 1, 1);

INSERT INTO BILETY VALUES(BILETY_SEQ.NEXTVAL, 5, 2, 30);

INSERT INTO REKLAMACJE VALUES(REKLAMACJE_SEQ.NEXTVAL, 'obsługa była niemila', 0, 1);

INSERT INTO REKLAMACJE VALUES(REKLAMACJE_SEQ.NEXTVAL, 'samolot nie poleciał', 1, 2);

INSERT INTO REKLAMACJE VALUES(REKLAMACJE_SEQ.NEXTVAL, 'zupa była za słona', 0, 3);

INSERT INTO REKLAMACJE VALUES(REKLAMACJE_SEQ.NEXTVAL, 'moj lot byl opozniony', 1, 3);

INSERT INTO REKLAMACJE VALUES(REKLAMACJE_SEQ.NEXTVAL, 'zgubiono mi bagaz', 0, 3);

```
INSERT INTO BAGAZE VALUES(BAGAZE_SEQ.NEXTVAL, NULL, default, 103, 21, 16, 1);
INSERT INTO BAGAZE VALUES(BAGAZE_SEQ.NEXTVAL, 15, default, 13, 33, 37, 3);
INSERT INTO BAGAZE VALUES(BAGAZE_SEQ.NEXTVAL, 5, default, 10, 10, 5, 4);
INSERT INTO BAGAZE VALUES(BAGAZE_SEQ.NEXTVAL, NULL, default, 4, 10, 15, 5);
INSERT INTO BAGAZE VALUES(BAGAZE_SEQ.NEXTVAL, 5, default, 3, 5, 11, 2);
```

```
INSERT INTO PRZYDZIALY VALUES (1,1);
INSERT INTO PRZYDZIALY VALUES (1,2);
INSERT INTO PRZYDZIALY VALUES (2,1);
INSERT INTO PRZYDZIALY VALUES (2,3);
INSERT INTO PRZYDZIALY VALUES (3,4);
```

e) Utworzenie tabeli logów

```
CREATE TABLE logi
(
    data_zmiany date,
    typ_zmiany CHAR,
    id_lotu_stare number(6,0),
    id_lotu_nowe number(6,0),
    stara_data date,
    nowa_data date,
    stara_bramka number(6,0),
    nowa_bramka number(6,0),
    stary_status char(2),
    nowy_status char(2),
```

```
stare_id_samolotu number(6,0),  
nowe_id_samolotu number(6,0),  
stare_id_trasy number(6,0),  
nowe_id_trasy number(6,0)  
);
```

e) Utworzenie triggera do wywoływania logów

```
CREATE OR REPLACE TRIGGER LOGGER AFTER INSERT OR UPDATE OR DELETE ON LOTY  
FOR EACH ROW  
DECLARE  
zmiana char(1);  
BEGIN  
IF UPDATING THEN  
zmiana := 'u';  
END IF;  
IF INSERTING THEN  
zmiana := 'i';  
END IF;  
IF DELETING THEN  
zmiana := 'd';  
END IF;  
  
insert into logi values(SYSDATE, zmiana, :old.id_lotu, :new.id_lotu, :old.data, :new.data, :old.bramka,  
                        :new.bramka, :old.status, :new.status, :old.id_samolotu,  
                        :new.id_samolotu, :old.id_trasy, :new.id_trasy);  
END;
```