

Wojskowa Akademia Techniczna
im. Jarosława Dąbrowskiego

Sprawozdanie z projektu semestralnego z przedmiotu:
Bazy danych (semestr zimowy 2021/2022)

Sprawozdanie z projektu o tytule:

Lotniska Europy

Wykonał: Radosław Relidzyński

Grupa: WCY20IY4S1

Data wykonania: 1.02.2022 r.

Spis treści	
Bazy danych (semestr zimowy 2021/2022).....	1
Treść zadania	2
Wprowadzenie.....	3
Założenia.....	3
Model logiczny.....	4
Model relacyjny	4
Instalacja bazy	5
Testowanie poprawności danych	5
Sprawdzanie poprawności lotów	5
Sprawdzanie poprawności licencji pilotów oraz ich lotów	5
Sprawdzanie zgodności samolotu na bilecie z samolotem na locie.....	6
Opis perspektyw	7
1) „KOSZTY_BILETOW”	7
2) „NAJWIECEJ_LOTOW”	7
3) „NAJCZESTSI_PILOCI”	8
4) „ZWYCIĘZCY_KONKURSU”	8
5) „ZYSK_SAMOLOTOW”	9
Listing programu.....	9
Skrypt tworzący tabele „create.sql”	9
Skrypt wypełniający tabele „data.sql”	13
Skrypt sprawdzający sumę wierszy tabel „check.sql”	26
Skrypt testujący poprawność wartości tabel „test.sql”	27
Skrypt tworzący perspektywy „views.sql”	28
Skrypt deinstalujący cały projekt „delete.sql”	29

Treść zadania

Cel:

Opracowanie projektu zaliczeniowego, na który składa się implementacja w środowisku Oracle modelu relacyjnej bazy danych oraz zestaw zdań SQL na podstawie, których można prezentować zawarte w bazie dane.

Harmonogram projektu:

1. Opracowanie logicznego modelu bazy danych, w oparciu o który realizowane będą dalsze części projektu.
2. Implementacja modelu relacyjnego na udostępnionym serwerze Oracle.
3. Opracowanie skryptów wdrożeniowych służących do implementacji i zarządzania projektem.
4. Opracowanie dokumentacji projektu.

Termin zakończenia projektu:

Termin zasadniczy - trzy dni przed ostatnimi zajęciami.

Szczegółowe założenia projektowe:

1. Opracować model bazy danych dowolnego fragmentu rzeczywistości zawierający elementy transakcyjne (sprzedaż, wypożyczenie, naprawa itp.). Dodatkowo strony transakcji podlegają wymiarowaniu poprzez tabele słownikowe (grupy produktowe, województwa, marki samochodów itp.). Do realizacji tego punktu należy wykorzystać narzędzie Oracle Data Modeler. Model powinien być zaimplementowany na serwerze Oracle 12c (city.wsisiz.edu.pl). Model logiczny powinien zawierać do 10 encji będących w związkach M:N (obowiązkowo) i 1:N, w którym występują atrybuty służące do liczenia (koszty, zyski, punkty itp.).
2. Do wszystkich tabel należy wprowadzić sensowne dane na podstawie skryptu zawierającego odpowiednie zdania SQL oraz sekwencje.
3. Opracować w postaci minimum czterech perspektyw wyniki zbiorcze obrazujące efekty działania zaprojektowanego oprogramowania na podstawie wprowadzonych danych. Perspektywy powinny pokazywać typowe wskaźniki biznesowe, na przykład zyski z prowadzonej działalności, wielkość sprzedaży poszczególnych produktów w funkcji czasu (lata, kwartały czy miesiące). Perspektywy powinny być zbudowane w sensowny sposób w oparciu o co najmniej trzy tabele. Można wykorzystać również zmienne wiązania symulując działanie w aplikacji.
4. Opracować skrypty wdrożeniowe umożliwiające instalację i deinstalację projektu na dowolnym koncie. Skrypt instalujący powinien zawierać zdania SQL tworzące obiekty bazodanowe (tabele, sekwencje, perspektywy) oraz zdania wprowadzające dane. Dopuszcza się możliwość istnienia kilku skryptów, na przykład tworzenie obiektów i wprowadzanie danych do tabel. Skrypt deinstalujący powinien usunąć ze schematu cały projekt.
5. W postaci zdań select zademonstrować użycie perspektyw, zdań podrzędnych nieskorelowanych i skorelowanych umieszczonych we frazach from, where i having zawierających funkcje agregujące oraz limitowanie wierszy (rownum).
6. Opracować dokumentację projektową zawierającą:
 - 6.1. Analizę biznesową projektowanej rzeczywistości,
 - 6.2. Model logiczny i relacyjny bazy danych,
 - 6.3. Oprogramowanie tworzące bazę danych,
 - 6.4. Skrypty wdrożeniowe instalujące i deinstalujące zrealizowany projekt,
 - 6.5. Instrukcję instalacji projektu i sprawdzenia jego poprawności,

Dodatkowe założenia (nieobowiązkowe):

7. Opracować programowy generator danych przy pomocy języka PL/SQL. Generator powinien w sposób automatyczny lub manualny generować transakcje na podstawie wprowadzonych danych stałych (strony transakcji i dane słownikowe). Zbudowany powinien być ze sparametryzowanych procedur i funkcji PL/SQL (ewentualnie zawierających zmienne wiązania). Do realizacji tego punktu w sposób automatyczny można wykorzystać obiekty programowe *jobs*.
8. Wykonać fragment aplikacji w Oracle Application Express. Wymaga to wcześniejszego uzgodnienia w celu przygotowania środowiska programowego oraz omówienia założeń na tworzoną aplikację.

Wprowadzenie

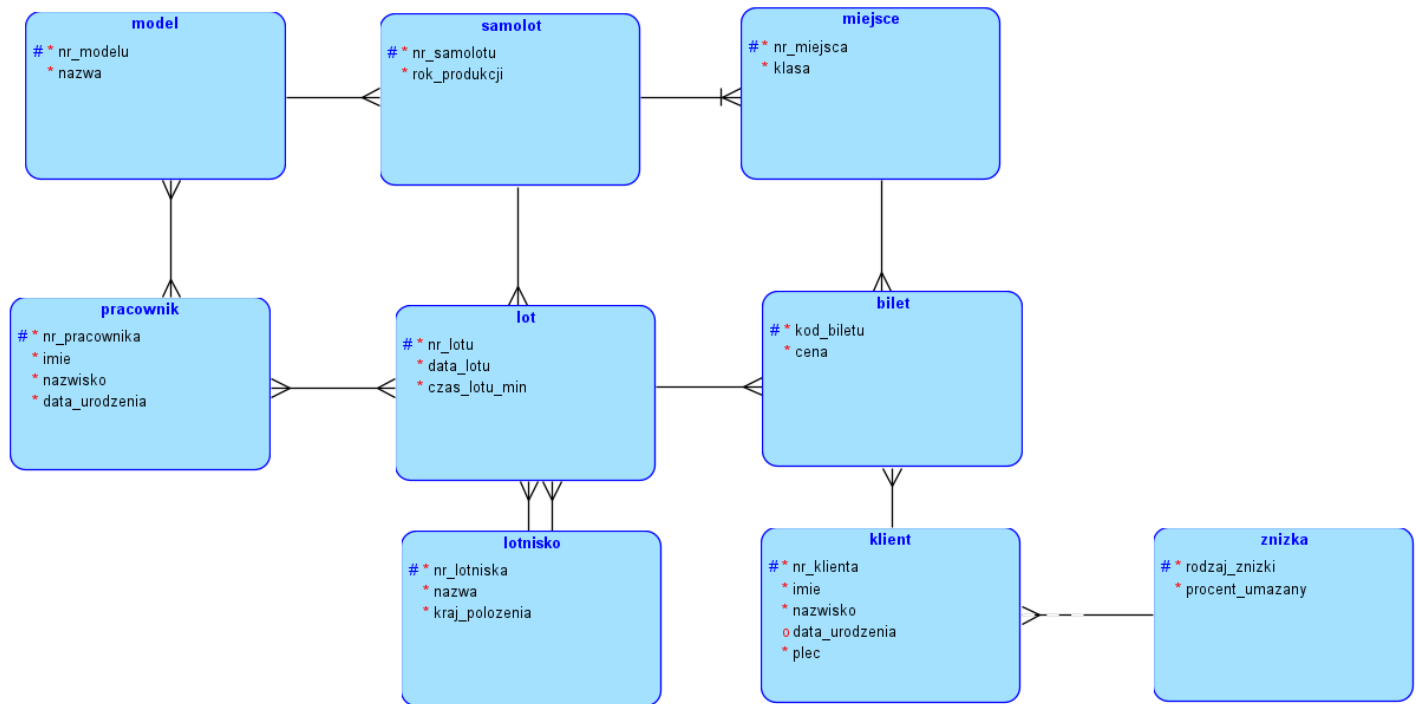
Baza danych stworzona w ramach projektu pełni zadanie zbierania i zarządzania informacjami dotyczącymi wybranych lotnisk Europy oraz lotów między nimi. W ramach tego powołane zostają poniższe encje:

- Samolot – informacje o samolocie
- Miejsce – informacje o miejscu w samolocie
- Model – informacje o modelach samolotów
- Pracownik – Informacje o pracownikach
- Licencja – intersekcja między pracownikiem i modelem, mówi, jakim samolotem może lecieć dany pracownik
- Lotnisko – informacje o lotniskach
- Lot – informacje o locie na podstawie samolotu i lotnisk
- Rola w locie – intersekcja między pracownikiem i lotem, mówi, jaką rolę pełni dany pracownik w danym locie
- Klient – informacje o kliencie
- Zniżki – informacje o zniżkach klientów
- Bilet – informacje o bilecie na podstawie lotu, miejsca w samolocie oraz klienta

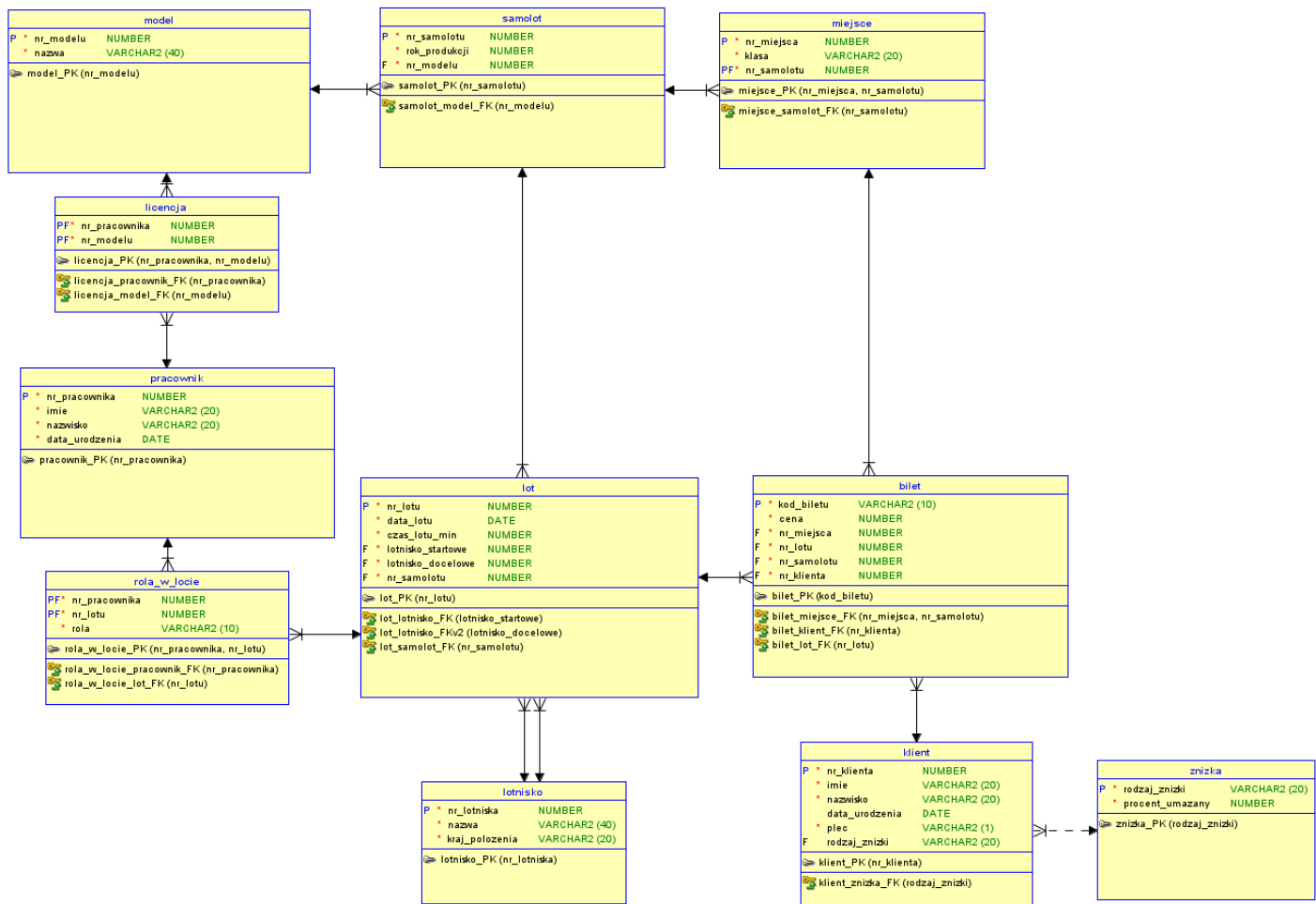
Założenia

- Baza ma możliwość przechowania dowolnej ilości danych pasujących do wymagań w encji
- Każdy atrybut został zadeklarowany z uwzględnieniem potrzebnego typu danych oraz konieczności uzupełniania pola
- Zastosowane dane są jednoznaczne i w sposób klarowny przedstawiają analizowaną rzeczywistość
- Stworzona baza danych daje możliwość do rozwijania jej bez konieczności przemodelowania

Model logiczny



Model relacyjny



Instalacja bazy

- 1) Tworzenie obiektów odbywa się przez wywołanie skryptu „create.sql”.
- 2) Uzupełnianie tabel odbywa się przez wywołanie skryptu „data.sql”.
- 3) W tym momencie baza została utworzona
- 4) Aby sprawdzić poprawność liczby wpisanych wierszy należy wywołać skrypt „check.sql”. Stan wierszy po wykonaniu instrukcji tworzenia i uzupełniania tabel powinien być następujący:

TABELA	LICZBA_WIERSZY
1 bilet	30
2 klient	10
3 licencja	12
4 lot	10
5 lotnisko	8
6 miejsce	50
7 model	10
8 pracownik	10
9 rola_w_locie	30
10 samolot	10
11 zniżka	3

- 5) Tworzenie widoków odbywa się przez wywołanie skryptu „views.sql”
- 6) W celu usunięcia wszystkich elementów z bazy danych należy wywołać skrypt „delete.sql”.

Testowanie poprawności danych

Wszystkie testy opierają się na zapytaniach, które w przypadku znalezienia błędu wypisują wiersze, w ramach których zaistniał błąd. Zapytania znajdują się w skrypcie „test.sql”. Skrypt możliwy jest do skopiowania ze sprawozdania (strona 27).

Sprawdzanie poprawności lotów

- Wymagania:
 - a) Czas lotu jest większy od zera
 - b) Lotnisko startowe jest różne od lotniska docelowego
- Instrukcja sprawdzająca:

```
--testowanie lotów
select l.nr_lotu bledne_loty
  from lot l
 where l.lotnisko_startowe = l.lotnisko_docelowe
 or l.czas_lotu_min <= 0
;
```

Sprawdzanie poprawności licencji pilotów oraz ich lotów

- Wymagania:
 - a) Pracownik, którego rolą podczas lotu jest bycie pilotem, musi posiadać licencję na samolot, który ten lot realizuje

- b) Zapytanie zwraca listę lotów, dla których pilot nie ma uprawnień do latania samolotem

- Instrukcja sprawdzająca:

```
--testowanie pilotów
with tmp as (
select l.nr_lotu dobre_loty
      from lot l
join samolot s on s.nr_samolotu = l.nr_samolotu
join rola_w_locie r on r.nr_lotu = l.nr_lotu
join pracownik p on p.nr_pracownika = r.nr_pracownika
join licencja x on x.nr_pracownika = p.nr_pracownika
join model m on m.nr_modelu = x.nr_modelu
              and m.nr_modelu = s.nr_modelu
order by l.nr_lotu
)
select l.nr_lotu bledne_loty
      from lot l
where l.nr_lotu not in (select tmp.dobre_loty from tmp)
;
```

Sprawdzanie zgodności samolotu na bilecie z samolotem na locie

- Wymagania:
 - a) Samolot przypisany do biletu musi być zgodnym z samolotem przypisanym do lotu, na który wypisany jest bilet

- Instrukcja sprawdzająca:

```
--testowanie samolotów
select b.kod_biletu bilet, b.nr_samolotu samolot_bilet, l.nr_samolotu
samolot_lot
      from bilet b
join miejsce m on b.nr_samolotu = m.nr_samolotu
join lot l on l.nr_lotu = b.nr_lotu and l.nr_samolotu <> b.nr_samolotu
;
```

Po wywołaniu skryptu „test.sql” powinny wyświetlić się 3 takie same komunikaty „no rows selected”.

Opis perspektyw

1) „KOSZTY_BILETOW”

Zadanie:

Zadaniem perspektywy jest wypisanie wszystkich biletów wraz z kosztem, jaki klient poniósł za bilet. Koszt obliczany jest jako cena pomnożona przez wyrażenie będące różnicą liczby 1 i liczby procent zniżki podzielonej przez 100 ($\text{koszt} = \text{cena} * (1 - \text{procent_umazany} / 100)$). Uzyskane wiersze mają być posortowane rosnąco względem kosztu.

Perspektywa:

```
--KOSZTY BILETÓW DLA PIERWSZEJ KLASY
create or replace view koszty_biletow (bilet, koszt)
as
    select b.kod_biletu bilet, round(b.cena*(1-
nvl(z.procent_umazany,0)/100),2) koszt from bilet b
    join klient k on k.nr_klienta = b.nr_klienta
    left join zniżka z on z.rodzaj_zniżki = k.rodzaj_zniżki
    join miejsce m on m.nr_miejsca = b.nr_miejsca and m.nr_samolotu =
b.nr_samolotu
    order by koszt desc
;
```

Widok perspektywy:

	BILET	KOSZT
1	EOM457	15,13
2	UJG957	17,02
3	YVY257	17,38
4	KZD756	30,39
5	HMF431	39,33
6	QRS931	44,26
7	FHQ738	49,77
8	GAL850	50,34
9	EQC031	54,29
10	RND132	58,06

(30 wierszy)

2) „NAJWIECEJ_LOTOW”

Zadanie:

Zadaniem perspektywy jest podanie największej liczby lotów, jakie odbył jeden pilot.

Perspektywa ta będzie potrzebna przy tworzeniu następnej.

Perspektywa:

```
--NAJWIĘKSZA ILOŚĆ LOTÓW JEDNEGO PILOTA
CREATE OR REPLACE VIEW najwiecej_lotow (najwiecej)
as
    select max(count(*)) najwiecej
    from pracownik p
```

```

join rola_w_locie r on r.nr_pracownika = p.nr_pracownika and r.rola =
'pilot'
join lot l on l.nr_lotu = r.nr_lotu
group by p.nr_pracownika

```

Widok perspektywy:

NAJWIECEJ	
1	3

3) „NAJCZESTSI_PILOCI”

Zadanie:

Zadaniem perspektywy jest wypisanie tych pilotów, którzy najczęściej byli pilotami podczas lotu. Jako pilot brana jest konkatenacja imienia i nazwiska pracownika. Do stworzenia tej perspektywy zostanie wykorzystana poprzednia zwracająca maksymalną ilość lotów jednego pilota.

Perspektywa:

```

--PILOCI O NAJWIEKSZEJ ILOSCI LOTOW
CREATE OR REPLACE VIEW najczestsi_piloci (pilot)
as
select p.imie||' '||p.nazwisko pilot
from pracownik p
where p.nr_pracownika in (
select p.nr_pracownika
from pracownik p
join rola_w_locie r on r.nr_pracownika = p.nr_pracownika and
r.rola = 'pilot'
group by p.nr_pracownika
having count(*) = (select najwiecej from najwiecej_lotow)
)
;

```

Widok perspektywy:

PILOT	
1	Piotr Gała
2	Waldemar Świerczewski

4) „ZWYCIĘZCY_KONKURSU”

Zadanie:

Ogłoszony zostaje konkurs, który mówi, że klienci, którzy zarezerwowali miejsce w samolocie o numerze równym reszty z dzielenia swojego roku urodzenia przez 5 wygrywają darmowy przelot ($\text{rok}(\text{data_urodzenia}) \% 5 = \text{numer miejsca}$). Zadaniem perspektywy jest wypisanie wszystkich zwycięzców konkursu. Konkurs obejmuje jedynie samoloty o modelu z serii „Boeing”.

Perspektywa:

```

--KLIENCI, KTÓRZY WYGRALI KONKURS

```



```
CREATE OR REPLACE VIEW zwyciezcy_konkursu (zwyciezca)
as
  select distinct k.imie||' '||k.nazwisko zwyciezca
  from klient k
  join bilet b on b.nr_klienta = k.nr_klienta
  join miejsce m on m.nr_samolotu = b.nr_samolotu and m.nr_miejsca =
b.nr_miejsca
  join samolot s on s.nr_samolotu = m.nr_samolotu
  join model x on x.nr_modelu = s.nr_modelu and x.nazwa like '%Boeing%'
  where mod(extract(year from k.data_urodzenia),5) = m.nr_miejsca
;
```

Widok perspektywy:

	ZWYCIEZCA
1	Tobiasz Ciszek

5) „ZYSK_SAMOLOTOW”

Zadanie:

Zadaniem perspektywy jest wypisanie 3 samolotów (numeru i modelu), których lot/loty przyniosły największy zysk. Zyskiem jest suma kosztów za bilety (obliczone w pierwszej perspektywie). Wybrane samoloty mają być posortowane względem zysku.

Perspektywa:

```
--SAMOLOTY O NAJWIĘKSZYM ZYSKU
CREATE OR REPLACE VIEW najwiekszy_zysk (samolot, zysk)
as
  select 'nr.'||s.nr_samolotu||': '||m.nazwa samolot, sum(k.koszt) zysk
  from samolot s
  join model m on s.nr_modelu = m.nr_modelu
  join lot l on l.nr_samolotu = s.nr_samolotu
  join bilet b on b.nr_lotu = l.nr_lotu
  join koszty_biletow k on b.kod_biletu = k.bilet
  group by s.nr_samolotu, m.nazwa
  order by sum(k.koszt) desc
  fetch next 3 rows only
;
```

Widok perspektywy:

	SAMOLOT	ZYSK
1	nr.4: Antonov AN-225 Mrija	459,52
2	nr.3: A380-800 British Airways	401,54
3	nr.2: Boeing 747-200	342,37

Listing programu

Skrypt tworzący tabele „create.sql”

```
CREATE TABLE bilet (
```

```

    kod_biletu  VARCHAR2(10) NOT NULL,
    cena        NUMBER NOT NULL,
    nr_miejsca  NUMBER NOT NULL,
    nr_lotu     NUMBER NOT NULL,
    nr_samolotu NUMBER NOT NULL,
    nr_klienta  NUMBER NOT NULL
);

ALTER TABLE bilet ADD CONSTRAINT bilet_pk PRIMARY KEY ( kod_biletu );

CREATE TABLE klient (
    nr_klienta  NUMBER NOT NULL,
    imie        VARCHAR2(20) NOT NULL,
    nazwisko    VARCHAR2(20) NOT NULL,
    data_urodzenia DATE,
    plec        VARCHAR2(1) NOT NULL,
    rodzaj_znizki VARCHAR2(20)
);

ALTER TABLE klient ADD CONSTRAINT klient_pk PRIMARY KEY ( nr_klienta );

CREATE TABLE licencja (
    nr_pracownika NUMBER NOT NULL,
    nr_modelu      NUMBER NOT NULL
);

ALTER TABLE licencja ADD CONSTRAINT licencja_pk PRIMARY KEY (
    nr_pracownika,
    nr_modelu
);

CREATE TABLE lot (
    nr_lotu        NUMBER NOT NULL,
    data_lotu      DATE NOT NULL,
    czas_lotu_min  NUMBER NOT NULL,
    lotnisko_startowe NUMBER NOT NULL,
    lotnisko_docelowe NUMBER NOT NULL,
    nr_samolotu    NUMBER NOT NULL
);

ALTER TABLE lot ADD CONSTRAINT lot_pk PRIMARY KEY ( nr_lotu );

CREATE TABLE lotnisko (
    nr_lotniska  NUMBER NOT NULL,
    nazwa        VARCHAR2(40) NOT NULL,
    kraj_polozenia VARCHAR2(20) NOT NULL
);

ALTER TABLE lotnisko ADD CONSTRAINT lotnisko_pk PRIMARY KEY ( nr_lotniska );

```

```

CREATE TABLE miejsce (
    nr_miejsca    NUMBER NOT NULL,
    klasa         VARCHAR2(20) NOT NULL,
    nr_samolotu   NUMBER NOT NULL
);

ALTER TABLE miejsce ADD CONSTRAINT miejsce_pk PRIMARY KEY ( nr_miejsca,
                                                            nr_samolotu
);

CREATE TABLE model (
    nr_modelu     NUMBER NOT NULL,
    nazwa         VARCHAR2(40) NOT NULL
);

ALTER TABLE model ADD CONSTRAINT model_pk PRIMARY KEY ( nr_modelu );

CREATE TABLE pracownik (
    nr_pracownika NUMBER NOT NULL,
    imie          VARCHAR2(20) NOT NULL,
    nazwisko      VARCHAR2(20) NOT NULL,
    data_urodzenia DATE NOT NULL
);

ALTER TABLE pracownik ADD CONSTRAINT pracownik_pk PRIMARY KEY (
nr_pracownika );

CREATE TABLE rola_w_locie (
    nr_pracownika NUMBER NOT NULL,
    nr_lotu        NUMBER NOT NULL,
    rola           VARCHAR2(10) NOT NULL
);

ALTER TABLE rola_w_locie ADD CONSTRAINT rola_w_locie_pk PRIMARY KEY (
nr_pracownika,
                                                                    nr_
lotu );

CREATE TABLE samolot (
    nr_samolotu   NUMBER NOT NULL,
    rok_produkcji NUMBER NOT NULL,
    nr_modelu     NUMBER NOT NULL
);

ALTER TABLE samolot ADD CONSTRAINT samolot_pk PRIMARY KEY ( nr_samolotu
);

CREATE TABLE znizka (
    rodzaj_znizki VARCHAR2(20) NOT NULL,
    procent_umazany NUMBER NOT NULL
);

```

```
ALTER TABLE znizka ADD CONSTRAINT znizka_pk PRIMARY KEY ( rodzaj_znizki );
```

```
ALTER TABLE bilet  
    ADD CONSTRAINT bilet_klient_fk FOREIGN KEY ( nr_klienta )  
        REFERENCES klient ( nr_klienta );
```

```
ALTER TABLE bilet  
    ADD CONSTRAINT bilet_lot_fk FOREIGN KEY ( nr_lotu )  
        REFERENCES lot ( nr_lotu );
```

```
ALTER TABLE bilet  
    ADD CONSTRAINT bilet_miejsce_fk FOREIGN KEY ( nr_miejsca,  
                                                nr_samolotu )  
        REFERENCES miejsce ( nr_miejsca,  
                             nr_samolotu );
```

```
ALTER TABLE klient  
    ADD CONSTRAINT klient_znizka_fk FOREIGN KEY ( rodzaj_znizki )  
        REFERENCES znizka ( rodzaj_znizki );
```

```
ALTER TABLE licencja  
    ADD CONSTRAINT licencja_model_fk FOREIGN KEY ( nr_modelu )  
        REFERENCES model ( nr_modelu );
```

```
ALTER TABLE licencja  
    ADD CONSTRAINT licencja_pracownik_fk FOREIGN KEY ( nr_pracownika )  
        REFERENCES pracownik ( nr_pracownika );
```

```
ALTER TABLE lot  
    ADD CONSTRAINT lot_lotnisko_fk FOREIGN KEY ( lotnisko_startowe )  
        REFERENCES lotnisko ( nr_lotniska );
```

```
ALTER TABLE lot  
    ADD CONSTRAINT lot_lotnisko_fkv2 FOREIGN KEY ( lotnisko_docelowe )  
        REFERENCES lotnisko ( nr_lotniska );
```

```
ALTER TABLE lot  
    ADD CONSTRAINT lot_samolot_fk FOREIGN KEY ( nr_samolotu )  
        REFERENCES samolot ( nr_samolotu );
```

```
ALTER TABLE miejsce  
    ADD CONSTRAINT miejsce_samolot_fk FOREIGN KEY ( nr_samolotu )  
        REFERENCES samolot ( nr_samolotu );
```

```
ALTER TABLE rola_w_locie  
    ADD CONSTRAINT rola_w_locie_lot_fk FOREIGN KEY ( nr_lotu )  
        REFERENCES lot ( nr_lotu );
```

```
ALTER TABLE rola_w_locie
```

```

        ADD CONSTRAINT rola_w_locie_pracownik_fk FOREIGN KEY ( nr_pracownika
    )
        REFERENCES pracownik ( nr_pracownika );

ALTER TABLE samolot
    ADD CONSTRAINT samolot_model_fk FOREIGN KEY ( nr_modelu )
        REFERENCES model ( nr_modelu );

```

Skrypt wypełniający tabelę „data.sql”

```

SET DEFINE OFF

--ZNIZKA

INSERT INTO ZNIZKA (RODZAJ_ZNIZKI, PROCENT_UMAZANY)
VALUES ('rodzinna', 37);

INSERT INTO ZNIZKA (RODZAJ_ZNIZKI, PROCENT_UMAZANY)
VALUES ('student', 50);

INSERT INTO ZNIZKA (RODZAJ_ZNIZKI, PROCENT_UMAZANY)
VALUES ('krwiodawca', 75);

--KLIENT

CREATE SEQUENCE seq;

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Kamil', 'Madaj', to_date('01/12/2000',
'MM/DD/RRRR'), 'student', 'M');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Emilia', 'Książkiewicz', to_date('06/02/1990',
'MM/DD/RRRR'), 'rodzinna', 'K');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Mateusz', 'Dybek', to_date('06/12/1993',
'MM/DD/RRRR'), NULL, 'M');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Zofia', 'Komosa', to_date('12/05/2001',
'MM/DD/RRRR'), 'student', 'K');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)

```

```

VALUES (seq.nextval, 'Krystyna', 'Czarniak', to_date('05/12/1992',
'MM/DD/RRRR'), NULL, 'K');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Janusz', 'Grzelczak', to_date('03/08/2002',
'MM/DD/RRRR'), 'student', 'M');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Albert', 'Świątkowski', to_date('06/04/1995',
'MM/DD/RRRR'), 'krwiodawca', 'M');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Krzysztof', 'Cieluch', to_date('09/10/1991',
'MM/DD/RRRR'), 'rodzinna', 'M');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Małgorzata', 'Handzlik', to_date('11/08/2001',
'MM/DD/RRRR'), NULL, 'K');

INSERT INTO KLIENT (NR_KLIENTA, IMIE, NAZWISKO, DATA_URODZENIA,
RODZAJ_ZNIZKI, PLEC)
VALUES (seq.nextval, 'Tobiasz', 'Ciszek', to_date('03/08/1996',
'MM/DD/RRRR'), 'rodzinna', 'M');

--LOTNISKO

ALTER SEQUENCE seq RESTART;

INSERT INTO LOTNISKO (NR_LOTNISKA, NAZWA, KRAJ_POLOZENIA)
VALUES (seq.nextval, 'Londyn Heathrow', 'Anglia');

INSERT INTO LOTNISKO (NR_LOTNISKA, NAZWA, KRAJ_POLOZENIA)
VALUES (seq.nextval, 'Paryż Charles de Gaulle', 'Francja');

INSERT INTO LOTNISKO (NR_LOTNISKA, NAZWA, KRAJ_POLOZENIA)
VALUES (seq.nextval, 'Frankfurt International Airport', 'Niemcy');

INSERT INTO LOTNISKO (NR_LOTNISKA, NAZWA, KRAJ_POLOZENIA)
VALUES (seq.nextval, 'Amsterdam Airport ', 'Holandia');

INSERT INTO LOTNISKO (NR_LOTNISKA, NAZWA, KRAJ_POLOZENIA)
VALUES (seq.nextval, 'Rzym Fiumicino', 'Włochy');

INSERT INTO LOTNISKO (NR_LOTNISKA, NAZWA, KRAJ_POLOZENIA)
VALUES (seq.nextval, 'Madryt', 'Hiszpania');

INSERT INTO LOTNISKO (NR_LOTNISKA, NAZWA, KRAJ_POLOZENIA)

```

```

VALUES (seq.nextval, 'Monachium', 'Niemcy');

INSERT INTO LOTNISKO (NR_LOTNISKA, NAZWA, KRAJ_POLOZENIA)
VALUES (seq.nextval, 'Barcelona Airport', 'Hiszpania');

--MODEL

ALTER SEQUENCE seq RESTART;

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'Boeing 787-8 Dreamliner');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'Boeing 747-200');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'DHC-6 Twin Otter');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'Antonov AN-225 Mrija');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'Boeing 777-300ER');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'DC-4 Balair');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'A380-800 British Airways');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'Concorde');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'Boeing 747-300');

INSERT INTO MODEL (NR_MODELU, NAZWA)
VALUES (seq.nextval, 'C-54D Skymaster');

--PRACOWNIK

ALTER SEQUENCE seq RESTART;

INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Mariusz', 'Szuszkiewicz', to_date('01/07/1985',
'MM/DD/RRRR'));

INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Piotr', 'Gała', to_date('01/25/1981',
'MM/DD/RRRR'));

```

```
INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Waldemar', 'Świerczewski', to_date('11/14/1992',
'MM/DD/RRRR'));
```

```
INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Maja', 'Kozik', to_date('11/15/1990',
'MM/DD/RRRR'));
```

```
INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Joanna', 'Krupa', to_date('07/13/1991',
'MM/DD/RRRR'));
```

```
INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Natalia', 'Kortas', to_date('03/05/1983',
'MM/DD/RRRR'));
```

```
INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Tomasz', 'Klasa', to_date('06/09/1981',
'MM/DD/RRRR'));
```

```
INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Szymon', 'Bucior', to_date('01/02/1989',
'MM/DD/RRRR'));
```

```
INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Emil', 'Mieczkowski', to_date('07/21/1982',
'MM/DD/RRRR'));
```

```
INSERT INTO PRACOWNIK (NR_PRACOWNIKA, IMIE, NAZWISKO, DATA_URODZENIA)
VALUES (seq.nextval, 'Maria', 'Wieczorek', to_date('05/22/1990',
'MM/DD/RRRR'));
```

```
--LICENCJA
```

```
INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (1, 3);
```

```
INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (1, 1);
```

```
INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (1, 4);
```

```
INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (2, 10);
```

```
INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (2, 1);
```

```
INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (2, 8);
```



```
INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (3, 7);

INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (3, 8);

INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (3, 3);

INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (4, 2);

INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (4, 8);

INSERT INTO LICENCJA (NR_PRACOWNIKA, NR_MODELU)
VALUES (4, 4);

--SAMOLOT

ALTER SEQUENCE seq RESTART;

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2013, 8);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2015, 2);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2010, 7);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2005, 4);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2008, 5);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2014, 1);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2019, 7);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2010, 10);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
VALUES (seq.nextval, 2013, 8);

INSERT INTO SAMOLOT (NR_SAMOLOTU, ROK_PRODUKCJI, NR_MODELU)
```

```

VALUES (seq.nextval, 2010, 1);

--LOT

ALTER SEQUENCE seq RESTART;

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('04/16/2019', 'MM/DD/RRRR'), 240, 3, 5, 1);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('06/23/2016', 'MM/DD/RRRR'), 270, 2, 8, 3);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('01/13/2019', 'MM/DD/RRRR'), 240, 10, 7, 2);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('05/31/2017', 'MM/DD/RRRR'), 120, 1, 6, 3);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('01/01/2015', 'MM/DD/RRRR'), 150, 8, 7, 5);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('09/26/2012', 'MM/DD/RRRR'), 240, 7, 1, 5);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('09/23/2015', 'MM/DD/RRRR'), 60, 3, 3, 4);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('03/29/2017', 'MM/DD/RRRR'), 90, 4, 6, 3);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('07/23/2010', 'MM/DD/RRRR'), 60, 2, 8, 7);

INSERT INTO LOT (NR_LOTU, DATA_LOTU, CZAS_LOTU_MIN, NR_SAMOLOTU,
LOTNISKO_STARTOWE, LOTNISKO_DOCELOWE)
VALUES (seq.nextval, to_date('03/19/2019', 'MM/DD/RRRR'), 90, 4, 2, 3);

--ROLA_W_LOCIE

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (1, 3, 'pilot');

```

```
INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (1, 5, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (1, 9, 'mechanik');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (2, 4, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (2, 6, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (2, 8, 'mechanik');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (3, 2, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (3, 8, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (3, 9, 'mechanik');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (4, 2, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (4, 6, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (4, 9, 'mechanik');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (5, 2, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (5, 7, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (5, 9, 'mechanik');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (6, 3, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (6, 7, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (6, 10, 'mechanik');
```

```
INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (7, 3, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (7, 7, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (7, 10, 'mechanik');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (8, 1, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (8, 6, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (8, 9, 'mechanik');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (9, 4, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (9, 5, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (9, 9, 'mechanik');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (10, 1, 'pilot');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (10, 8, 'steward');

INSERT INTO ROLA_W_LOCIE (NR_LOTU, NR_PRACOWNIKA, ROLA)
VALUES (10, 10, 'mechanik');

--MIEJSCE

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 1);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 1);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 1);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 1);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
```

```
VALUES (5, 'ekonomiczna', 1);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 2);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 2);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 2);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 2);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 2);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 3);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 3);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 3);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 3);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 3);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 4);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 4);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 4);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 4);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 4);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 5);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
```

```
VALUES (2, 'pierwsza', 5);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 5);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 5);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 5);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 6);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 6);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 6);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 6);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 6);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 7);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 7);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 7);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 7);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 7);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 8);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 8);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 8);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
```

```

VALUES (4, 'ekonomiczna', 8);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 8);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 9);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 9);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 9);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 9);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 9);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (1, 'biznes', 10);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (2, 'pierwsza', 10);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (3, 'pierwsza', 10);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (4, 'ekonomiczna', 10);

INSERT INTO MIEJSCE (NR_MIEJSCA, KLASA, NR_SAMOLOTU)
VALUES (5, 'ekonomiczna', 10);

--BILET

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('NZY114', 1, 3, 1, 4, 141.78);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('RND132', 1, 3, 2, 2, 92.16);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('YMD153', 1, 3, 4, 3, 70.89);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)

```

```
VALUES ('NIE210', 2, 2, 1, 10, 139.0);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('BIM239', 2, 2, 3, 9, 90.35);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('VYV257', 2, 2, 5, 7, 69.5);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('QZS315', 3, 10, 1, 5, 144.36);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('HHC332', 3, 10, 3, 2, 93.83);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('IBP359', 3, 10, 5, 9, 72.18);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('PFW415', 4, 1, 1, 5, 121.02);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('HMF431', 4, 1, 3, 1, 78.66);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('EOM457', 4, 1, 4, 7, 60.51);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('SGD512', 5, 8, 1, 2, 142.76);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('LEJ535', 5, 8, 3, 5, 92.79);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('WTX553', 5, 8, 4, 3, 71.38);

INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,
NR_KLIENTA, CENA)
VALUES ('GOQ610', 6, 7, 1, 10, 160.18);
```



```
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('GGR638', 6, 7, 3, 8, 104.12);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('WBP655', 6, 7, 5, 5, 80.09);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('WEZ715', 7, 3, 1, 5, 121.54);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('FHQ738', 7, 3, 3, 8, 79.0);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('KZD756', 7, 3, 5, 6, 60.77);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('IVV812', 8, 4, 1, 2, 159.82);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('DEX838', 8, 4, 3, 8, 103.88);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('GAL850', 8, 4, 4, 10, 79.91);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('IJE910', 9, 2, 1, 10, 136.18);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('QRS931', 9, 2, 3, 1, 88.52);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('UJG957', 9, 2, 5, 7, 68.09);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('DTE018', 10, 4, 1, 8, 167.04);  
  
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('EQC031', 10, 4, 3, 1, 108.58);
```

```
INSERT INTO BILET (KOD_BILETU, NR_LOTU, NR_SAMOLOTU, NR_MIEJSCA,  
NR_KLIENTA, CENA)  
VALUES ('YRA053', 10, 4, 5, 3, 83.52);  
  
DROP SEQUENCE seq;
```

Skrypt sprawdzający sumę wierszy tabel „check.sql”

```
select 'bilet' Tabela, count(*) liczba_wierszy  
from bilet  
union  
select 'klient', count(*)  
from klient  
union  
select 'licencja', count(*)  
from licencja  
union  
select 'lot', count(*)  
from lot  
union  
select 'lotnisko', count(*)  
from lotnisko  
union  
select 'miejsce', count(*)  
from miejsce  
union  
select 'model', count(*)  
from model  
union  
select 'pracownik', count(*)  
from pracownik  
union  
select 'rola_w_locie', count(*)  
from rola_w_locie  
union  
select 'samolot', count(*)  
from samolot  
union  
select 'znizka', count(*)  
from znizka
```

Skrypt testujący poprawność wartości tabel „test.sql”

```
--testowanie lotów
select l.nr_lotu bledne_loty
      from lot l
where l.lotnisko_startowe = l.lotnisko_docelowe
or l.czas_lotu_min <= 0
;

--testowanie pilotów
with tmp as (
select l.nr_lotu dobre_loty
      from lot l
join samolot s on s.nr_samolotu = l.nr_samolotu
join rola_w_locie r on r.nr_lotu = l.nr_lotu
join pracownik p on p.nr_pracownika = r.nr_pracownika
join licencja x on x.nr_pracownika = p.nr_pracownika
join model m on m.nr_modelu = x.nr_modelu
      and m.nr_modelu = s.nr_modelu
order by l.nr_lotu
)
select l.nr_lotu bledne_loty
      from lot l
where l.nr_lotu not in (select tmp.dobre_loty from tmp)
;

--testowanie samolotów
select b.kod_biletu bilet, b.nr_samolotu samolot_bilet, l.nr_samolotu
samolot_lot
      from bilet b
join miejsce m on b.nr_samolotu = m.nr_samolotu
join lot l on l.nr_lotu = b.nr_lotu and l.nr_samolotu <> b.nr_samolotu
;
```

Skrypt tworzący perspektywy „views.sql”

```
--KOSZTY BILETÓW DLA PIERWSZEJ KLASY
create or replace view koszty_biletow (bilet, koszt)
as
    select b.kod_biletu bilet, round(b.cena*(1-
nvl(z.procent_umazany,0)/100),2) koszt from bilet b
    join klient k on k.nr_klienta = b.nr_klienta
    left join zniżka z on z.rodzaj_zniżki = k.rodzaj_zniżki
    join miejsce m on m.nr_miejsca = b.nr_miejsca and m.nr_samolotu =
b.nr_samolotu
    order by koszt
;

--NAJWIĘKSZA ILOŚĆ LOTÓW JEDNEGO PILOTA
CREATE OR REPLACE VIEW najwiecej_lotow (najwiecej)
as
    select max(count(*)) najwiecej
    from pracownik p
    join rola_w_locie r on r.nr_pracownika = p.nr_pracownika and r.rola =
'pilot'
    join lot l on l.nr_lotu = r.nr_lotu
    group by p.nr_pracownika
;

--PILOCI O NAJWIĘKSZEJ ILOSCI LOTOW
CREATE OR REPLACE VIEW najczestsi_piloci (pilot)
as
    select p.imie||' '||p.nazwisko pilot
    from pracownik p
    where p.nr_pracownika in (
        select p.nr_pracownika
        from pracownik p
        join rola_w_locie r on r.nr_pracownika = p.nr_pracownika and
r.rola = 'pilot'
        group by p.nr_pracownika
        having count(*) = (select najwiecej from najwiecej_lotow)
    )
;

--KLIENCI, KTÓRZY WYGRALI KONKURS
CREATE OR REPLACE VIEW zwyciezcy_konkursu (zwyciezca)
as
    select distinct k.imie||' '||k.nazwisko zwyciezca
    from klient k
    join bilet b on b.nr_klienta = k.nr_klienta
    join miejsce m on m.nr_samolotu = b.nr_samolotu and m.nr_miejsca =
b.nr_miejsca
    join samolot s on s.nr_samolotu = m.nr_samolotu
    join model x on x.nr_modelu = s.nr_modelu and x.nazwa like '%Boeing%'
    where mod(extract(year from k.data_urodzenia),5) = m.nr_miejsca
;

--SAMOLOTY O NAJWIĘKSZYM ZYSKU
```

```

CREATE OR REPLACE VIEW najwiekszy_zysk (samolot, zysk)
as
    select 'nr.'||s.nr_samolotu||': '||m.nazwa samolot, sum(k.koszt) zysk
    from samolot s
    join model m on s.nr_modelu = m.nr_modelu
    join lot l on l.nr_samolotu = s.nr_samolotu
    join bilet b on b.nr_lotu = l.nr_lotu
    join koszty_biletow k on b.kod_biletu = k.bilet
    group by s.nr_samolotu, m.nazwa
    order by sum(k.koszt) desc
    fetch next 3 rows only
;

```

Skrypt deinstalujący cały projekt „delete.sql”

```

DROP TABLE licencja;
DROP TABLE rola_w_locie;
DROP TABLE bilet;
DROP TABLE klient;
DROP TABLE miejsce;
DROP TABLE pracownik;
DROP TABLE lot;
DROP TABLE znizka;
DROP TABLE lotnisko;
DROP TABLE samolot;
DROP TABLE model;
DROP VIEW koszty_biletow;
DROP VIEW najwiecej_lotow;
DROP VIEW najczestsi_piloci;
DROP VIEW zwyciezcy_konkursu;
DROP VIEW najwiekszy_zysk;

```