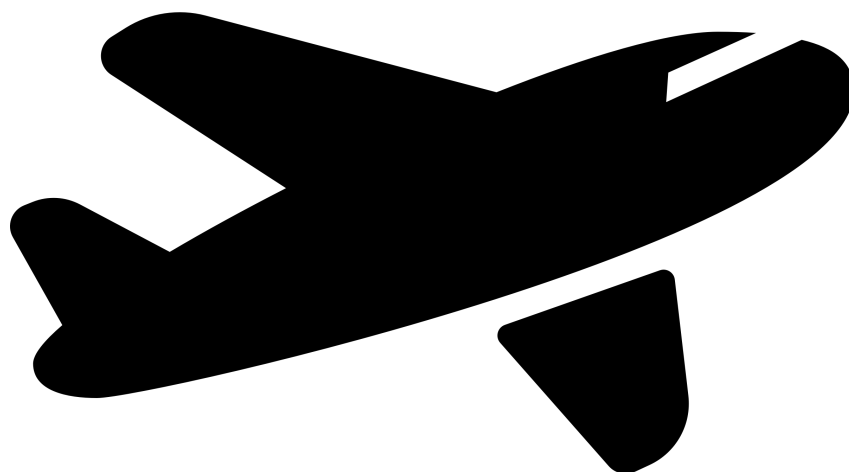


SKYBASE MANAGER 2017

OPIS BAZY DANYCH



Autor: Sebastian Zabrzyski
Data publikacji: 1 maj 2017

SPIS TREŚCI

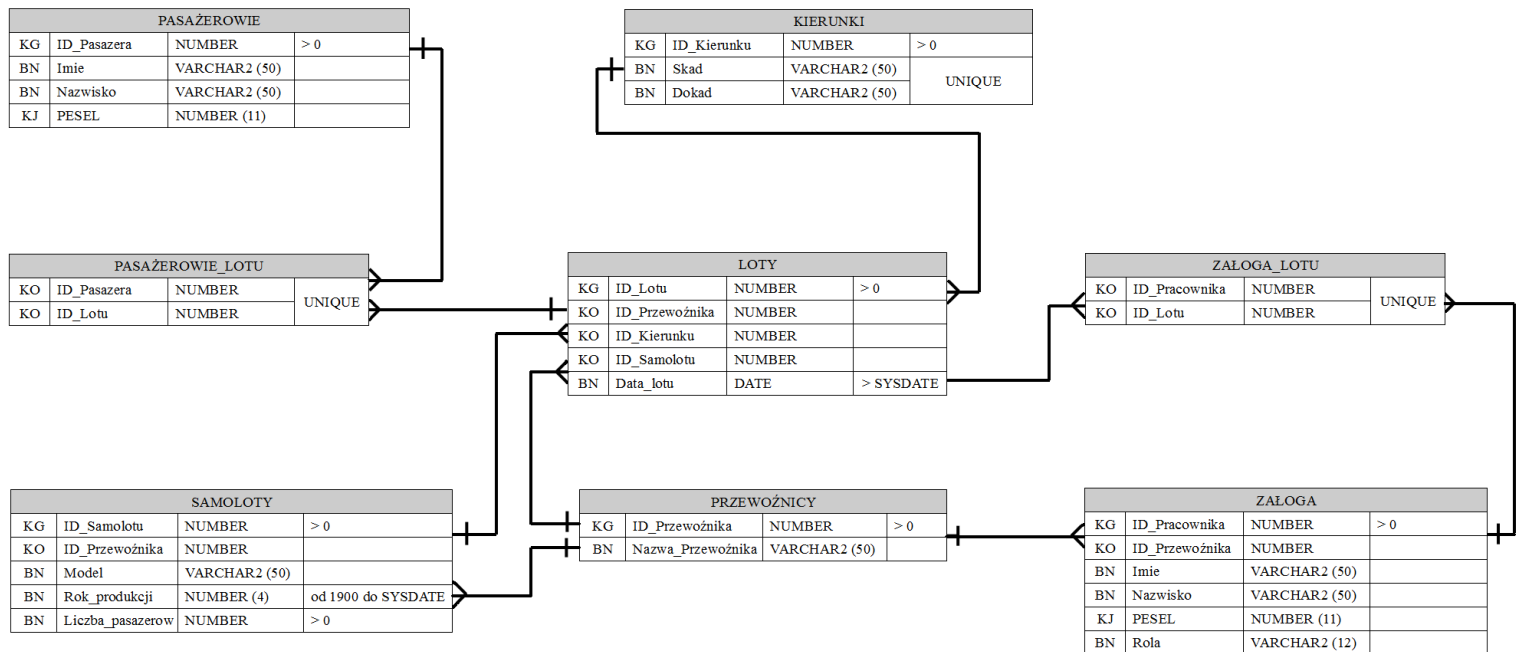
1. Wstęp.....	2
2. Diagram ERD.....	2
3. Tabele.....	3
4. Relacje.....	4
5. Połączenie z bazą danych.....	5
6. Skrypt SQL.....	6
7. Procedury SQL.....	10

1. Wstęp

Opisywana baza danych została stworzona w systemie relacyjnym Oracle Database z wykorzystaniem języka SQL oraz PL/SQL i jest umieszczona na serwerze lokalnym Użytkownika. Baza przeznaczona jest do obsługi portu lotniczego, umożliwiając zarządzanie danymi dotyczącymi lotów pasażerskich.

2. Diagram ERD

Diagram ERD bazy danych przedstawia się następująco:



Oznaczenia:

KG -Klucz główny

KO – Klucz obcy

KJ – Klucz jednoznaczny

3. Tabele

Baza składa się z 8 tabel omówionych poniżej:

- Pasazerowie

Tabela zawiera dane dotyczące pasażerów korzystających z usług portu lotniczego:

1. ID_Pasazera
2. Imie
3. Nazwisko
4. PESEL

- Załoga

Tabela zawiera dane dotyczące załogi biorącej udział w lotach pasażerskich:

1. ID_Pracownika
2. ID_Przewoźnika
3. Imie
4. Nazwisko
5. PESEL
6. Rola (Kapitan/Drugi pilot/Stewardessa/Steward)

- Kierunki

Tabela zawiera dane o kierunkach połączeń lotniczych:

1. ID_Kierunku
2. Skad
3. Dokad

- Samoloty

Tabela zawiera dane o samolotach wykorzystywanych w przewozie:

1. ID_Samolotu
2. ID_Przewoźnika
3. Model

- 4. Rok_produkcji
- 5. Liczba_pasazerow

- Loty

Tabela zawiera dane o historii lotów pasażerskich:

- 1. ID_Lotu
- 2. ID_Przewoznika
- 3. ID_Kierunku
- 4. ID_Samolotui
- 5. Data_lotu (w formacie YYYY-MM-DD)

- Pasazerowie_lotu

Tabela zawiera dane o pasażerach konkretnego lotu:

- 1. ID_Pasazera
- 2. ID_Lotu

- Zaloga_lotu

Tabela zawiera dane o załodze konkretnego lotu:

- 1. ID_Pracownika
- 2. ID_Lotu

- Przewoźnicy

Tabela zawiera dane o przewoźnikach świadczących usługi lotów pasażerskich:

- 1. ID_Przewoznika
- 2. Nazwa_przewoznika

4. Relacje

Tabela główna → Tabela podporządkowana:

PASAŻEROWIE → PASAŻEROWIE_LOT

PRACOWNICY → PRACOWNICY_LOT

SAMOLOTY → LOTY

LOTY → PRACOWNICY_LOT
LOTY → PASAŻEROWIE_LOT
KIERUNKI → LOTY
PRZEWOŹNICY → LOTY
PRZEWOŹNICY → SAMOLOTY
PRZEWOŹNICY → PRACOWNICY

Przy próbie modyfikacji klucza głównego w tabeli głównej zostanie obrane podejście kaskadowe.

Przy próbie usunięcia wiersza w tabeli głównej zostanie obrane podejście standardowe.

4. Połączenie z bazą danych

Aby połączyć się z bazą danych na serwerze lokalnym, należy podać następujące dane:

Hostname: **localhost**

Port: **1521**

SID: **oracle**

Username: **oracle**

Password: **oracle**

New / Select Database Connection

Connection Name	Connection Details
orade	orade@//localhost:1521:oracle

Connection Name: orade
Username: orade
Password:
☐ Save Password

Oracle Access

Connection Type: Basic Role: default

Hostname: localhost
Port: 1521
☒ SID: oracle
☐ Service name:
☐ OS Authentication ☐ Kerberos Authentication ☐ Proxy Connection

Status :
Pomoc Save Clear Test Connect Anuluj

5. Skrypt SQL

Do utworzenia bazy danych wykorzystano następujący skrypt SQL:

```
1 CREATE TABLE Pasazerowie(  
2 ID_Pasazera NUMBER NOT NULL PRIMARY KEY,  
3 Imię VARCHAR2(50) NOT NULL ,  
4 Nazwisko VARCHAR2(50) NOT NULL ,  
5 Pesel NUMBER(11) NOT NULL UNIQUE  
6 );  
7  
8 commit;  
9  
10 CREATE TABLE Zaloga(  
11 ID_Pracownika NUMBER NOT NULL PRIMARY KEY,  
12 ID_Przewoźnika NUMBER NOT NULL,  
13 Imię VARCHAR2(50) NOT NULL ,  
14 Nazwisko VARCHAR2(50) NOT NULL ,  
15 Pesel NUMBER(11) NOT NULL UNIQUE,  
16 Rola VARCHAR2(11) NOT NULL  
17 );  
18  
19 commit;  
20  
21 CREATE TABLE Kierunki(  
22 ID_Kierunku NUMBER NOT NULL PRIMARY KEY,  
23 Skąd VARCHAR2(50) NOT NULL ,  
24 Dokąd VARCHAR2(50) NOT NULL  
25 );  
26  
27 commit;  
28  
29 CREATE TABLE Samoloty(  
30 ID_Samolotu NUMBER NOT NULL PRIMARY KEY,  
31 ID_Przewoźnika NUMBER NOT NULL,  
32 Model VARCHAR2(50) NOT NULL ,  
33 Rok_produkcji NUMBER(4) NOT NULL ,  
34 Liczba_pasazerow NUMBER NOT NULL  
35 );  
36  
37 commit;  
38  
39 CREATE TABLE Loty(  
40 ID_Lotu NUMBER NOT NULL PRIMARY KEY,  
41 ID_Przewoźnika NUMBER NOT NULL,  
42 ID_Kierunku NUMBER NOT NULL ,  
43 ID_Samolotu NUMBER NOT NULL ,  
44 Data_lotu DATE NOT NULL  
45 );  
46  
47 commit;  
48  
49 CREATE TABLE Pasazerowie_lotu(  
50 ID_Pasazera NUMBER NOT NULL ,  
51 ID_Lotu NUMBER NOT NULL  
52 );  
53  
54 commit;  
55  
56 CREATE TABLE Zaloga_lotu(  
57 ID_Pracownika NUMBER NOT NULL ,  
58 ID_Lotu NUMBER NOT NULL  
59 );  
60  
61 commit;  
62  
63 CREATE TABLE Przewoźnicy(  
64 ID_Przewoźnika NUMBER NOT NULL PRIMARY KEY,
```

```

65 Nazwa_Przewoznika VARCHAR2(50) NOT NULL
66 );
67
68 commit;
69
70 ALTER TABLE LOTY ADD CONSTRAINT fk_idkierunku FOREIGN KEY (ID_Kierunku) REFERENCES
KIERUNKI(ID_Kierunku);
71 ALTER TABLE LOTY ADD CONSTRAINT fk_idsamolotu FOREIGN KEY (ID_Samolotu) REFERENCES
SAMOLOTY(ID_Samolotu);
72 ALTER TABLE LOTY ADD CONSTRAINT fk_idprzewoznika FOREIGN KEY (ID_Przewoznika)
REFERENCES PRZEWOZNICY(ID_Przewoznika);
73 ALTER TABLE Zaloga ADD CONSTRAINT fk_idprzewoznika2 FOREIGN KEY (ID_Przewoznika)
REFERENCES PRZEWOZNICY(ID_Przewoznika);
74 ALTER TABLE SAMOLOTY ADD CONSTRAINT fk_idprzewoznika3 FOREIGN KEY (ID_Przewoznika)
REFERENCES PRZEWOZNICY(ID_Przewoznika);
75 ALTER TABLE Pasazerowie_lotu ADD CONSTRAINT fk_idpasazazera FOREIGN KEY (ID_Pasazera)
REFERENCES PASAZEROWIE(ID_Pasazera);
76 ALTER TABLE Pasazerowie_lotu ADD CONSTRAINT fk_idlotu FOREIGN KEY (ID_Lotu) REFERENCES
LOTY(ID_Lotu);
77 ALTER TABLE Zaloga_lotu ADD CONSTRAINT fk_idlotu2 FOREIGN KEY (ID_Lotu) REFERENCES LOTY
(ID_Lotu);
78 ALTER TABLE Zaloga_lotu ADD CONSTRAINT fk_idpracownika FOREIGN KEY (ID_Pracownika)
REFERENCES Zaloga(ID_Pracownika);
79
80
81 ALTER TABLE Pasazerowie ADD CHECK (Pesel > 9999999999);
82 ALTER TABLE Zaloga ADD CHECK (Pesel > 9999999999);
83 ALTER TABLE Samoloty ADD CHECK (Rok_produkcji > 1900);
84 ALTER TABLE Samoloty ADD CHECK (ID_Samolotu > 0);
85 ALTER TABLE Przewoznicy ADD CHECK (ID_Przewoznika > 0);
86 ALTER TABLE Samoloty ADD CHECK (Liczba_pasazerow > 0);
87 ALTER TABLE Pasazerowie ADD CHECK (ID_Pasazera > 0);
88 ALTER TABLE Loty ADD CHECK (ID_Lotu > 0);
89 ALTER TABLE Kierunki ADD CHECK (ID_Kierunku > 0);
90 ALTER TABLE Zaloga ADD CHECK (ID_Pracownika > 0);
91
92 ALTER TABLE Pasazerowie_lotu ADD CONSTRAINT uq_Pasazerowie_lotu UNIQUE(ID_Pasazera,
ID_Lotu);
93 ALTER TABLE Przewoznicy ADD CONSTRAINT uq_Przewoznicy UNIQUE(ID_Przewoznika,
Nazwa_Przewoznika);
94 ALTER TABLE Zaloga_lotu ADD CONSTRAINT uq_Zaloga_lotu UNIQUE(ID_Pracownika, ID_Lotu);
95 ALTER TABLE Kierunki ADD CONSTRAINT uq_Kierunki UNIQUE(Skad, Dokad, ID_Kierunku);
96
97 commit;
98
99 CREATE OR REPLACE TRIGGER ID_Pasazera_update
100 AFTER UPDATE OF ID_Pasazera ON Pasazerowie FOR EACH ROW
101 BEGIN
102 UPDATE Pasazerowie_lotu
103 SET ID_Pasazera = :new.ID_Pasazera
104 WHERE ID_Pasazera = :old.ID_Pasazera;
105 END;
106 /
107
108 CREATE OR REPLACE TRIGGER ID_Lotu_update
109 AFTER UPDATE OF ID_Lotu ON Loty FOR EACH ROW
110 BEGIN
111
112 UPDATE Pasazerowie_lotu
113 SET ID_Lotu = :new.ID_Lotu
114 WHERE ID_Lotu = :old.ID_Lotu;
115
116 UPDATE Zaloga_lotu
117 SET ID_Lotu = :new.ID_Lotu
118 WHERE ID_Lotu = :old.ID_Lotu;
119
120 END;
121 /
122
123 CREATE OR REPLACE TRIGGER ID_Pracownika_update

```

```

124 BEFORE UPDATE OF ID_Pracownika ON Zaloga FOR EACH ROW
125 BEGIN
126 UPDATE Zaloga_lotu
127 SET ID_Pracownika = :new.ID_Pracownika
128 WHERE ID_Pracownika = :old.ID_Pracownika;
129 END;
130 /
131
132 CREATE OR REPLACE TRIGGER ID_Samolotu_update
133 AFTER UPDATE OF ID_Samolotu ON Samoloty FOR EACH ROW
134 BEGIN
135 UPDATE Loty
136 SET ID_Samolotu = :new.ID_Samolotu
137 WHERE ID_Samolotu = :old.ID_Samolotu;
138 END;
139 /
140
141 CREATE OR REPLACE TRIGGER ID_Kierunku_update
142 AFTER UPDATE OF ID_Kierunku ON Kierunki FOR EACH ROW
143 BEGIN
144 UPDATE Loty
145 SET ID_Kierunku = :new.ID_Kierunku
146 WHERE ID_Kierunku = :old.ID_Kierunku;
147 END;
148 /
149
150 CREATE OR REPLACE TRIGGER ID_Przewoznika_update
151 AFTER UPDATE OF ID_Przewoznika ON Przewoznicy FOR EACH ROW
152 BEGIN
153
154 UPDATE Loty
155 SET ID_Przewoznika= :new.ID_Przewoznika
156 WHERE ID_Przewoznika = :old.ID_Przewoznika;
157
158 UPDATE Zaloga
159 SET ID_Przewoznika= :new.ID_Przewoznika
160 WHERE ID_Przewoznika = :old.ID_Przewoznika;
161
162
163 UPDATE Samoloty
164 SET ID_Przewoznika= :new.ID_Przewoznika
165 WHERE ID_Przewoznika = :old.ID_Przewoznika;
166
167 END;
168 /
169
170 CREATE OR REPLACE TRIGGER loty_check_insert
171 AFTER INSERT ON Loty
172 FOR EACH ROW
173 DECLARE
174 ID_Przewoznika1 NUMBER;
175 BEGIN
176 SELECT ID_Przewoznika
177 INTO ID_Przewoznika1
178 FROM Samoloty
179 WHERE ID_Samolotu = :new.ID_Samolotu;
180 IF ID_Przewoznika1 != :new.ID_Przewoznika THEN
181 RAISE_APPLICATION_ERROR (-20000, 'Samolot nie należy do tego przewoźnika');
182 END IF;
183
184 IF :new.Data_lotu < TRUNC(sysdate) THEN
185 RAISE_APPLICATION_ERROR (-20000, 'Niepoprawna data lotu');
186 END IF;
187
188 END;
189 /
190
191
192 CREATE OR REPLACE TRIGGER loty_check_update
193 AFTER UPDATE ON Loty

```



```

194 FOR EACH ROW
195 BEGIN
196
197 IF :new.Data_lotu < '1900-01-01' THEN
198 RAISE_APPLICATION_ERROR (-20000, 'Niepoprawna data lotu');
199 END IF;
200
201 END;
202 /
203
204
205 CREATE OR REPLACE TRIGGER samoloty_check_insertupdate
206 AFTER INSERT OR UPDATE ON Samoloty
207 FOR EACH ROW
208 BEGIN
209
210 IF :new.Rok_produkcji < 1900 OR :new.Rok_produkcji > EXTRACT(YEAR FROM sysdate)
211 THEN
212 RAISE_APPLICATION_ERROR (-20000, 'Niepoprawny rok produkcji');
213 END IF;
214 END;
215 /
216
217
218
219 CREATE OR REPLACE TRIGGER zaloga_check_insert
220 AFTER INSERT ON Zaloga
221 FOR EACH ROW
222 BEGIN
223 IF :new.Rola != 'Kapitan' AND :new.Rola != 'Drugi pilot' AND :new.Rola !=
224 'Stewardessa' AND :new.Rola != 'Steward' THEN
225 RAISE_APPLICATION_ERROR (-20000, 'Niepoprawna rola członka załogi');
226 END IF;
227 END;
228 /
229
230
231
232
233 CREATE OR REPLACE TRIGGER Zaloga_lotu_check_insert
234 BEFORE INSERT ON Zaloga_lotu
235 FOR EACH ROW
236 DECLARE
237 ID_Przewoznika1 NUMBER;
238 ID_Przewoznika2 NUMBER;
239 Rola1 VARCHAR2(11);
240 Rola2 VARCHAR2(11);
241 Rola3 VARCHAR2(11);
242 Ilosc NUMBER;
243 BEGIN
244
245 SELECT ID_Przewoznika
246 INTO ID_Przewoznika1
247 FROM Loty
248 WHERE ID_Lotu = :new.ID_Lotu;
249
250 SELECT ID_Przewoznika
251 INTO ID_Przewoznika2
252 FROM Zaloga
253 WHERE ID_Pracownika = :new.ID_Pracownika;
254
255 IF ID_Przewoznika2 != ID_Przewoznika1 THEN
256 RAISE_APPLICATION_ERROR (-20000, 'Pracownik nie jest zatrudniony u tego
przewoźnika');
257 END IF;
258
259 SELECT Rola
260 INTO Rola1
261 FROM Zaloga

```

```

262 WHERE ID_Pracownika = :new.ID_Pracownika;
263
264 IF Rola1 = 'Kapitan' OR Rola1 = 'Drugi pilot' THEN
265
266
267
268 FOR rec IN (SELECT ID_Pracownika FROM Zaloga_lotu WHERE ID_Lotu = :new.ID_Lotu)
269 LOOP
270 SELECT Rola INTO Rola2 FROM Zaloga WHERE ID_Pracownika = rec.ID_Pracownika;
271
272 IF Rola2 = Rola1 THEN
273 RAISE_APPLICATION_ERROR (-20000, 'Istnieje juz wybrany kapitan/drugi pilot
dla tego lotu');
274 END IF;
275
276
277 END LOOP;
278 END IF;
279
280
281
282 END;
283 /

```

7. Procedury SQL

Baza danych zawiera kilka podprogramów typu wyzwalacz, które reagują na określone zdarzenia podczas dodawania lub modyfikacji jej zawartości:

- W razie modyfikacji ID_Pasazera w tabeli Pasazerowie zmiane ulega również ID_Pasazera w tabeli Pasazerowie_lotu.
- W razie modyfikacji ID_Pracownika w tabeli Zaloga zmiane ulega również ID_Pracownika w tabeli Zaloga_lotu.
- W razie modyfikacji ID_Lotu w tabeli Loty zmiane ulega również ID_Lotu w tabelach Zaloga_lotu oraz Pasazerowie_lotu.
- W razie modyfikacji ID_Samolotu w tabeli Samoloty zmiane ulega również ID_Samolotu w tabeli Loty.
- W razie modyfikacji ID_Kierunku w tabeli Kierunki zmiane ulega również ID_Kierunkow tabeli Loty.
- W razie modyfikacji ID_Przewoźnika w tabeli Przewoźnicy zmiane ulega również ID_Przewoźnika w tabeli Loty, Zaloga oraz Samoloty.
- W razie wstawienia nowego rekordu do tabeli Loty sprawdzane jest, czy samolot należy do podanego przewoźnika oraz czy data lotu jest równa lub późniejsza niż obecna data.
- W razie wstawienia nowego rekordu do tabeli Samoloty lub modyfikacji istniejącego sprawdzane jest, czy data produkcji samolotu nie jest zbyt wczesna.
- W razie wstawienia nowego rekordu do tabeli Zaloga sprawdzane jest, czy podana rola członka załogi jest dozwolona.
- W razie wstawienia nowego rekordu do tabeli Zaloga_lotu sprawdzane jest, czy członek załogi pracuje u danego przewoźnika oraz czy w locie bierze udział tylko jeden kapitan/drugi pilot.