

Laboratorium 1: Oracle PL/SQL

Uladzislau Tumilovich

1 Tabele

1.1 Osoby

```
1 — TABLE OSOBY
2 CREATE TABLE OSOBY
3 (
4     ID_OSOBY INT GENERATED ALWAYS AS IDENTITY NOT NULL
5     , IMIE VARCHAR2(50)
6     , NAZWISKO VARCHAR2(50)
7     , PESEL VARCHAR2(11)
8     , KONTAKT VARCHAR2(100)
9     , CONSTRAINT OSOBY_PK PRIMARY KEY
10    (
11        ID_OSOBY
12    )
13    ENABLE
14 );
```

1.2 Wycieczki

```
1 — TABLE WYCIECZKI
2 CREATE TABLE WYCIECZKI
3 (
4     ID_WYCIECZKI INT GENERATED ALWAYS AS IDENTITY NOT NULL
5     , NAZWA VARCHAR2(100)
6     , KRAJ VARCHAR2(50)
7     , DATA DATE
8     , OPIS VARCHAR2(200)
9     , LICZBA_MIEJSC INT
10    , CONSTRAINT WYCIECZKI_PK PRIMARY KEY
11    (
12        ID_WYCIECZKI
13    )
14    ENABLE
15 );
```

1.3 Rezerwacje

```
1 — TABLE REZERWACJE
2 CREATE TABLE REZERWACJE
3 (
4     NR_REZERWACJI INT GENERATED ALWAYS AS IDENTITY NOT NULL
5     , ID_WYCIECZKI INT
6     , ID_OSOBY INT
7     , STATUS CHAR(1)
8     , CONSTRAINT REZERWACJE_PK PRIMARY KEY
9     (
10        NR_REZERWACJI
11    )
12    ENABLE
13 );
```

1.4 Constrainty

```
1  — CONSTRAINTS
2
3  ALTER TABLE REZERWACJE
4      ADD CONSTRAINT REZERWACJE_FK1 FOREIGN KEY
5      (
6          ID_OSOBY
7      )
8  REFERENCES OSOBY
9      (
10         ID_OSOBY
11     )
12  ENABLE;
13
14 ALTER TABLE REZERWACJE
15     ADD CONSTRAINT REZERWACJE_FK2 FOREIGN KEY
16     (
17         ID_WYCIECZKI
18     )
19 REFERENCES WYCIECZKI
20     (
21         ID_WYCIECZKI
22     )
23  ENABLE;
24
25 ALTER TABLE REZERWACJE
26     ADD CONSTRAINT REZERWACJE_CHK1 CHECK
27     (status IN ('N', 'P', 'Z', 'A'))
28  ENABLE;
```

2 Przykładowe dane

```
1  — INSERTING INITIAL DATA INTO TABLES
2
3  — TABLE: OSOBY
4  INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
5  VALUES ('Adam', 'Kowalski', '87654321', 'tel: 6623');
6
7  INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
8  VALUES ('Michal', 'Blumberg', '56652320', 'e-mail: blumberg@gmail.com');
9
10 INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
11 VALUES ('Kasia', 'Nowak', '73950372', 'tel: 73943');
12
13 INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
14 VALUES ('Luda', 'Ptuszkina', '74925927', 'tel: 34839');
15
16 INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
17 VALUES ('Piotr', 'Krusiak', '74922848', 'e-mail: piotr.krusiak@gmail.com');
18
19 INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
20 VALUES ('Slawik', 'Marlek', '92832732', 'tel: 39483');
21
22 INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
23 VALUES ('Agata', 'Lumsik', '57393483', 'tel: 2324');
24
25 INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
```

```

26 VALUES ( 'Flora', 'Gusiak', '48394839', 'e-mail: flora@mail.com' );
27
28 INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
29 VALUES ( 'Bartosz', 'Remilek', '34938493', 'tel: 23492' );
30
31 INSERT INTO OSOBY (IMIE, NAZWISKO, PESEL, KONTAKT)
32 VALUES ( 'Swiatlana', 'Luzik', '439384934', 'tel: 3243554' );
33
34
35 — TABLE: WYCIECZKI
36 INSERT INTO WYCIECZKI (NAZWA, KRAJ, DATA, OPIS, LICZBA_MIEJSC)
37 VALUES ( 'Wycieczka do Paryza', 'Francja', TO_DATE( '2016-01-01', 'YYYY-MM-DD' ), '
    Ciekawa wycieczka ... ', 3 );
38
39 INSERT INTO wycieczki (NAZWA, KRAJ, DATA, OPIS, LICZBA_MIEJSC)
40 VALUES ( 'Piekny Krakow', 'Polska', TO_DATE( '2017-02-03', 'YYYY-MM-DD' ), 'Najciekawa
    wycieczka ... ', 2 );
41
42 INSERT INTO wycieczki (NAZWA, KRAJ, DATA, OPIS, LICZBA_MIEJSC)
43 VALUES ( 'Wieliczka', 'Polska', TO_DATE( '2017-03-03', 'YYYY-MM-DD' ), 'Zadziwiajaca
    kopalnia ... ', 2 );
44
45 INSERT INTO wycieczki (NAZWA, KRAJ, DATA, OPIS, LICZBA_MIEJSC)
46 VALUES ( 'Piekne gory', 'Polska', TO_DATE( '2020-05-12', 'YYYY-MM-DD' ), 'Wysokie gory
    ... ', 3 );
47
48 — TABLE REZERWACJE
49 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
50 VALUES (3,1, 'N' );
51
52 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
53 VALUES (3,4, 'P' );
54
55 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
56 VALUES (3,5, 'Z' );
57
58 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
59 VALUES (4,6, 'P' );
60
61 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
62 VALUES (4,7, 'A' );
63
64 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
65 VALUES (5,8, 'Z' );
66
67 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
68 VALUES (5,9, 'N' );
69
70 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
71 VALUES (6,10, 'A' );
72
73 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
74 VALUES (6,12, 'P' );
75
76 INSERT INTO REZERWACJE(ID_WYCIECZKI, ID_OSOBY, STATUS)
77 VALUES (6,1, 'Z' );

```

3 Widoki

3.1 Rezerwacje wszystkie

```
1 — VIEW REZERWACJE WSZYSTKIE
2 CREATE OR REPLACE VIEW view_rezerwacje_wszystkie
3     AS
4     SELECT
5         r.NR_REZERWACJI,
6         w.ID_WYCIECZKI,
7         w.NAZWA,
8         w.KRAJ,
9         w.DATA,
10        o.IMIE,
11        o.NAZWISKO,
12        r.STATUS
13 FROM WYCIECZKI w
14 JOIN REZERWACJE r ON w.ID_WYCIECZKI = r.ID_WYCIECZKI
15 JOIN OSOBY o ON r.ID_OSOBY = o.ID_OSOBY;
```

3.2 Rezerwacje potwierdzone

```
1 — VIEW REZERWACJE_POTWIERDZONE
2 CREATE OR REPLACE VIEW view_rezerwacje_potwierdzone
3     AS
4     SELECT *
5     FROM view_rezerwacje_wszystkie rw
6     WHERE rw.STATUS = 'P' OR rw.STATUS = 'Z';
```

3.3 Rezerwacje w przyszłości

```
1 — VIEW REZERWACJE W PRZYSZŁOŚCI
2 CREATE OR REPLACE VIEW view_rezerwacje_w_przyszlosci
3     AS
4     SELECT *
5     FROM view_rezerwacje_wszystkie rw
6     WHERE rw.DATA > CURRENT_DATE;
```

3.4 Wycieczki miejsca

```
1  — VIEW WYCIECZKI Miejsca
2  CREATE OR REPLACE VIEW view_wycieczki_miejsca
3      AS
4      SELECT
5          w.ID_WYCIECZKI,
6          w.KRAJ,
7          w.DATA,
8          w.NAZWA,
9          w.LICZBA_MIEJSC,
10         (w.LICZBA_MIEJSC - (
11             SELECT COUNT(*)
12             FROM REZERWACJE r
13             WHERE r.ID_WYCIECZKI = w.ID_WYCIECZKI
14                 AND r.STATUS <> 'A'
15         )) AS LICZBA_WOLNYCH_MIEJSC
16     FROM WYCIECZKI w;
```

3.5 Wycieczki w przyszłości

```
1  — VIEW WYCIECZKI W PRZYSZŁOŚCI
2  CREATE OR REPLACE VIEW view_wycieczki_w_przyszlosci
3      AS
4      SELECT *
5      FROM view_wycieczki_miejsca wm
6      WHERE wm.DATA > CURRENT_DATE;
```

3.6 Wycieczki dostępne

```
1  — VIEW WYCIECZKI DOSTĘPNE
2  CREATE OR REPLACE VIEW view_wycieczki_dostepne
3      AS
4      SELECT *
5      FROM view_wycieczki_w_przyszlosci wm
6      WHERE wm.LICZBA_WOLNYCH_MIEJSC > 0;
```

3.7 Wycieczki osoby

```
1  — VIEW WYCIECZKI OSOBY
2  CREATE OR REPLACE VIEW view_wycieczki_osoby
3      AS
4      SELECT
5          o.ID_OSOBY,
6          o.IMIE,
7          o.NAZWISKO,
8          o.KONTAKT,
9          o.PESEL,
10         w.NAZWA,
11         w.KRAJ,
12         w.DATA,
13         r.STATUS
14     FROM OSOBY o
15         JOIN REZERWACJE r ON o.ID_OSOBY = r.ID_OSOBY
16         JOIN WYCIECZKI w ON r.ID_WYCIECZKI = w.ID_WYCIECZKI;
```

4 Obiekty

4.1 Wycieczka info

```
1  — TYPE WYCIECZKA_INFO
2  CREATE OR REPLACE TYPE type_wycieczka_info
3      AS OBJECT (
4      ID_WYCIECZKI      INT,
5      NAZWA              VARCHAR2(50) ,
6      KRAJ              VARCHAR2(100) ,
7      "DATA"            DATE,
8      IMIE              VARCHAR2(50) ,
9      NAZWISKO          VARCHAR2(50) ,
10     STATUS            CHAR(1)
11 );
12
13 — TYPE WYCIECZKA_INFO_TABLE
14 CREATE OR REPLACE TYPE type_wycieczka_info_table
15     IS TABLE OF type_wycieczka_info;
```

4.2 Osoba info

```
1  — TYPE OSOBA_INFO
2  CREATE OR REPLACE TYPE type_osoba_info
3      AS OBJECT (
4      ID_OSOBY          INT,
5      IMIE              VARCHAR2(50) ,
6      NAZWISKO          VARCHAR2(50) ,
7      KONTAKT           VARCHAR2(100) ,
8      PESEL             VARCHAR2(11) ,
9      NAZWA              VARCHAR2(50) ,
10     KRAJ              VARCHAR2(100) ,
11     "DATA"            DATE,
12     STATUS            CHAR(1)
13 );
14
15 — TYPE OSOBA_INFO_TABLE
16 CREATE OR REPLACE TYPE type_osoba_info_table
17     IS TABLE OF type_osoba_info;
```

4.3 Wycieczka dostępna

```
1  — TYPE WYCIECZKA_DOSTEPNA
2  CREATE OR REPLACE TYPE type_wycieczka_dostepna
3      AS OBJECT (
4      ID_WYCIECZKI      INT,
5      KRAJ              VARCHAR2(100) ,
6      "DATA"            DATE,
7      NAZWA              VARCHAR2(50) ,
8      LICZBA_MIEJSC     INT,
9      LICZBA_WOLNYCH_MIEJSC INT
10 );
11
12 — TYPE WYCIECZKA_DOSTEPNA_TABLE
13 CREATE OR REPLACE TYPE type_wycieczka_dostepna_table
14     IS TABLE OF type_wycieczka_dostepna;
```

5 Funkcje pobierające dane

Funkcje pobierające dane korzystają z obiektów

5.1 Uczestnicy wycieczki

```
1 — FUNCTION UCZESTNICY_WYCIECZKI
2 CREATE OR REPLACE
3 FUNCTION fnc_uczestnicy_wycieczki(id_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE)
4     RETURN type_wycieczka_info_table AS result_table type_wycieczka_info_table;
5     is_table INTEGER;
6     BEGIN
7         SELECT COUNT(*) INTO is_table
8         FROM WYCIECZKI w
9         WHERE w.ID_WYCIECZKI = fnc_uczestnicy_wycieczki.id_wycieczki;
10
11         IF is_table = 0 THEN
12             raise_application_error(-20014, 'Brak wycieczki o podanym id');
13         END IF;
14
15         SELECT type_wycieczka_info(rw.ID_WYCIECZKI, rw.NAZWA, rw.KRAJ, rw.DATA,
16     rw.IMIE,
17     rw.NAZWISKO, rw.STATUS)
18         BULK COLLECT INTO result_table
19         FROM VIEW_REZERWACJE_WSZYSTKIE rw
20         WHERE rw.ID_WYCIECZKI = fnc_uczestnicy_wycieczki.id_wycieczki
21         AND rw.STATUS <> 'A';
22
23     RETURN result_table;
24 END fnc_uczestnicy_wycieczki;
```

5.2 Rezerwacje osoby

```
1 — FUNCTION REZERWACJE_OSOBY
2 CREATE OR REPLACE
3 FUNCTION fnc_rezerwacje_osoby(id_osoby OSOBY.ID_OSOBY%TYPE)
4     RETURN type_osoba_info_table AS result_table type_osoba_info_table;
5     is_table INTEGER;
6     BEGIN
7         SELECT COUNT(*) INTO is_table
8         FROM OSOBY w
9         WHERE w.ID_OSOBY = fnc_rezerwacje_osoby.id_osoby;
10
11         IF is_table = 0 THEN
12             raise_application_error(-20014, 'Brak osoby o podanym id');
13         END IF;
14
15         SELECT type_osoba_info(wo.ID_OSOBY, wo.IMIE, wo.NAZWISKO, wo.KONTAKT, wo
16     .PESEL,
17     wo.NAZWA, wo.KRAJ, wo.DATA, wo.STATUS)
18         BULK COLLECT INTO result_table
19         FROM VIEW_WYCIECZKI_OSOBY wo
20         WHERE wo.ID_OSOBY = fnc_rezerwacje_osoby.id_osoby;
21
22     RETURN result_table;
23 END fnc_rezerwacje_osoby;
```


5.3 Dostępne wycieczki

```
1  — FUNCTION DOSTEPNE WYCIECZKI
2  CREATE OR REPLACE
3  FUNCTION fnc_dostepne_wycieczki(kraj WYCIECZKI.KRAJ%TYPE,
4                                  data_od WYCIECZKI.DATA%TYPE,
5                                  data_do WYCIECZKI.DATA%TYPE)
6  RETURN type_wycieczka_dostepna_table AS result_table
7  type_wycieczka_dostepna_table;
8  BEGIN
9      IF data_od < CURRENT_DATE THEN
10         raise_application_error(-20014, 'Data od nie moze byc wczesniej niz
11         dzisiejsza data');
12     END IF;
13     IF data_od > data_do THEN
14         raise_application_error(-20014, 'Data do nie moze byc wczesniej niz
15         data od');
16     END IF;
17     SELECT type_wycieczka_dostepna(wd.ID_WYCIECZKI, wd.KRAJ, wd.DATA, wd.
18     NAZWA, wd.LICZBA_MIEJSC,
19     wd.LICZBA_WOLNYCH_MIEJSC)
20     BULK COLLECT INTO result_table
21     FROM VIEW_WYCIECZKI_DOSTEPNE wd
22     WHERE wd.KRAJ = fnc_dostepne_wycieczki.kraj AND data_od <= wd.DATA
23     AND wd.DATA <= data_do;
24     RETURN result_table;
25 END fnc_dostepne_wycieczki;
```

6 Procedury modyfikujące dane

Procedury zostały odrazu dopasowane do tabeli REZERWACJE_LOG

6.1 Dodaj rezerwacje

```
1  — PROCEDURE DODAJ_REZERWACJE
2  CREATE OR REPLACE
3  PROCEDURE prc_dodaj_rezerwacje
4      (id_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE, id_osoby OSOBY.ID_OSOBY%TYPE) AS
5      is_table          INTEGER;
6      new_reservation_id INTEGER;
7  BEGIN
8      SELECT COUNT(*) INTO is_table
9      FROM OSOBY o
10     WHERE o.ID_OSOBY = prc_dodaj_rezerwacje.id_osoby;
11
12     IF is_table = 0 THEN
13         raise_application_error(-20014, 'Brak osoby o podanym id');
14     END IF;
15
16     SELECT COUNT(*) INTO is_table
17     FROM VIEW_WYCIECZKI_DOSTEPNE wd
18     WHERE wd.ID_WYCIECZKI = prc_dodaj_rezerwacje.id_wycieczki;
19
20     IF is_table = 0 THEN
21         raise_application_error(-20014, 'Brak dostępnej wycieczki o podanym id
22     ');
23     END IF;
24
25     SELECT COUNT(*) INTO is_table
26     FROM REZERWACJE r
27     WHERE r.ID_WYCIECZKI = prc_dodaj_rezerwacje.id_wycieczki
28           AND r.ID_OSOBY = prc_dodaj_rezerwacje.id_osoby;
29
30     IF is_table > 0 THEN
31         raise_application_error(-20014, 'Rezerwacja już istnieje');
32     END IF;
33
34     INSERT INTO REZERWACJE r (r.ID_WYCIECZKI, r.ID_OSOBY, r.STATUS)
35     VALUES (id_wycieczki, id_osoby, 'N')
36     RETURNING r.NR_REZERWACJI INTO new_reservation_id;
37
38     INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
39     VALUES (new_reservation_id, CURRENT_DATE, 'N');
```

6.2 Zmien status rezerwacji

```
1  — PROCEDURE ZMIEN_STATUS_REZERWACJI
2  CREATE OR REPLACE
3  PROCEDURE prc_zmien_status_rezerwacji
4      (nr_rezerwacji REZERWACJE.NR_REZERWACJI%TYPE, status REZERWACJE.STATUS%TYPE)
5      AS
6      is_table      INTEGER;
7      past_status   REZERWACJE.STATUS%TYPE;
8      BEGIN
9          SELECT COUNT(*) INTO is_table
10         FROM REZERWACJE r
11        WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji.nr_rezerwacji;
12
13         IF is_table = 0 THEN
14             raise_application_error(-20014, 'Brak rezerwacji o podanym nr');
15         END IF;
16
17         SELECT COUNT(*) INTO is_table
18        FROM VIEW_REZERWACJE_W_PRZYSZLOSCI rwp
19       WHERE rwp.NR_REZERWACJI = prc_zmien_status_rezerwacji.nr_rezerwacji;
20
21         IF is_table = 0 THEN
22             raise_application_error(-20014, 'Status rezerwacji mozna zmieniac
23 tylko ' ||
24                                     'dla wycieczek przyszlosci');
25         END IF;
26
27         SELECT COUNT(*) INTO is_table
28        FROM VIEW_WYCIEZKI_DOSTEPNE wd
29       JOIN REZERWACJE r ON r.ID_WYCIEZKI = wd.ID_WYCIEZKI
30      WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji.nr_rezerwacji;
31
32         SELECT r.STATUS INTO past_status
33        FROM REZERWACJE r
34       WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji.nr_rezerwacji;
35
36         IF prc_zmien_status_rezerwacji.past_status = 'A' AND is_table = 0 THEN
37             raise_application_error(-20014, 'Brak dostepnych miejsc');
38         END IF;
39
40         IF prc_zmien_status_rezerwacji.past_status = prc_zmien_status_rezerwacji
41 .status THEN
42             raise_application_error(-20014, 'Nie mozna zamienic status na taki
43 sam');
44         END IF;
45
46         UPDATE REZERWACJE r
47        SET STATUS = prc_zmien_status_rezerwacji.status
48       WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji.nr_rezerwacji;
49
50         INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
51        VALUES (prc_zmien_status_rezerwacji.nr_rezerwacji, CURRENT_DATE,
52                prc_zmien_status_rezerwacji.status);
53     END prc_zmien_status_rezerwacji;
```

6.3 Zmien liczbe miejsc

```
1  — PROCEDURE ZMIEN LICZBE MIEJSC
2  CREATE OR REPLACE
3  PROCEDURE prc_zmien_liczbe_miejsc
4      (id_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE, liczba_miejsc WYCIECZKI.
      LICZBA_MIEJSC%TYPE) AS
5      is_table      INTEGER;
6      places        INTEGER;
7      free_places   INTEGER;
8      BEGIN
9          SELECT COUNT(*) INTO is_table
10         FROM WYCIECZKI w
11         WHERE w.ID_WYCIECZKI = prc_zmien_liczbe_miejsc.id_wycieczki;
12
13         IF is_table = 0 THEN
14             raise_application_error(-20014, 'Nie istnieje wycieczki o
podanym id');
15         END IF;
16
17         SELECT COUNT(*) INTO is_table
18         FROM VIEW_WYCIECZKI_W_PRZYSZLOSCI wwp
19         WHERE wwp.ID_WYCIECZKI = prc_zmien_liczbe_miejsc.id_wycieczki;
20
21         IF is_table = 0 THEN
22             raise_application_error(-20014, 'Nie mozna zmieniac liczbe
miejsc skonczonej' ||
23                                     ' wycieczki');
24         END IF;
25
26         SELECT wd.LICZBA_MIEJSC INTO places
27         FROM VIEW_WYCIECZKI_DOSTEPNE wd
28         WHERE wd.ID_WYCIECZKI = prc_zmien_liczbe_miejsc.id_wycieczki;
29
30         IF prc_zmien_liczbe_miejsc.liczba_miejsc < places THEN
31
32             SELECT wd.LICZBA_WOLNYCH_MIEJSC INTO free_places
33             FROM VIEW_WYCIECZKI_DOSTEPNE wd
34             WHERE wd.ID_WYCIECZKI = prc_zmien_liczbe_miejsc.id_wycieczki;
35
36             IF places - free_places > prc_zmien_liczbe_miejsc.liczba_miejsc THEN
37                 raise_application_error(-20014, 'Nie mozna zmienic liczbe
miejsc na mniejsza' ||
38                                     ' niz zarejestrowanych
uczestnikow');
39             END IF;
40         END IF;
41
42         UPDATE WYCIECZKI
43         SET LICZBA_MIEJSC = prc_zmien_liczbe_miejsc.liczba_miejsc
44         WHERE ID_WYCIECZKI = prc_zmien_liczbe_miejsc.id_wycieczki;
45     END prc_zmien_liczbe_miejsc;
```

7 Dodanie tabeli REZERWACJE_LOG

7.1 Dodanie tabeli

```
1  — TABLE REZERWACJE_LOG
2  CREATE TABLE REZERWACJE_LOG
3  (
4      ID INT GENERATED ALWAYS AS IDENTITY NOT NULL
5      , ID_REZERWACJI INT
6      , DATA DATE
7      , STATUS CHAR(1)
8      , CONSTRAINT REZERWACJE_LOG_PK PRIMARY KEY
9      (
10         ID
11     )
12     ENABLE
13 );
```

7.2 Dodanie constraint'a

```
1  ALTER TABLE REZERWACJE_LOG
2      ADD CONSTRAINT
3      REZERWACJE_LOG_FK1 FOREIGN KEY
4      (
5          ID_REZERWACJI
6      )
7      REFERENCES REZERWACJE
8      (
9          NR_REZERWACJI
10     )
11     ENABLE;
```

8 Dodanie pola liczba_wolnych_miejsc do tabeli wycieczki

8.1 Dodane pole

```
1  ALTER TABLE WYCIECZKI
2      ADD LICZBA_WOLNYCH_MIEJSC INT;
```

8.2 Procedura update'ująca tabelę wycieczki do nowej formy

```
1  — PROCEDURE PRZELICZ_WOLNE_MIEJSCA
2  CREATE OR REPLACE
3  PROCEDURE prc_przelicz_wolne_miejsc
4      AS
5      BEGIN
6          UPDATE WYCIECZKI w
7          SET w.LICZBA_WOLNYCH_MIEJSC = w.LICZBA_MIEJSC -
8              (SELECT COUNT(*) FROM REZERWACJE r
9               WHERE r.ID_WYCIECZKI = w.ID_WYCIECZKI
10                  AND r.STATUS <> 'A');
11      END;
```

9 Zmienione widoki

9.1 Wycieczki miejsca 2

```
1 — VIEW WYCIECZKI_MIEJSCA2
2 CREATE OR REPLACE VIEW view_wycieczki_miejsca2
3 AS
4     SELECT
5         w.ID_WYCIECZKI,
6         w.KRAJ,
7         w.DATA,
8         w.NAZWA,
9         w.LICZBA_MIEJSC,
10        w.LICZBA_WOLNYCH_MIEJSC
11 FROM WYCIECZKI w;
```

9.2 Wycieczki W przyszłości 2

```
1 — VIEW WYCIECZKI_W_PRZYSZLOSCI2
2 CREATE OR REPLACE VIEW view_wycieczki_w_przyszlosci2
3 AS
4     SELECT *
5     FROM view_wycieczki_miejsca2 wm
6     WHERE wm.DATA > CURRENT_DATE;
```

9.3 Wycieczki dostępne 2

```
1 — VIEW WYCIECZKI_DOSTEPNE2
2 CREATE OR REPLACE VIEW view_wycieczki_dostepne2
3 AS
4     SELECT *
5     FROM view_wycieczki_w_przyszlosci2 wm
6     WHERE wm.LICZBA_WOLNYCH_MIEJSC > 0;
```

10 Zmienione funkcji

10.1 Dostępne wycieczki 2

```
1  — FUNCTION DOSTEPNE WYCIECZKI2
2  CREATE OR REPLACE
3  FUNCTION fnc_dostepne_wycieczki2(kraj WYCIECZKI.KRAJ%TYPE,
4                                     data_od WYCIECZKI.DATA%TYPE,
5                                     data_do WYCIECZKI.DATA%TYPE)
6  RETURN type_wycieczka_dostepna_table AS result_table
7  type_wycieczka_dostepna_table;
8  BEGIN
9      IF data_od < CURRENT_DATE THEN
10         raise_application_error(-20014, 'Data od nie moze byc wczesniej niz
11         dzisiejsza data');
12     END IF;
13
14     IF data_od > data_do THEN
15         raise_application_error(-20014, 'Data do nie moze byc wczesniej niz
16         data od');
17     END IF;
18
19     SELECT type_wycieczka_dostepna(wd.ID_WYCIECZKI, wd.KRAJ, wd.DATA, wd.
20     NAZWA, wd.LICZBA_MIEJSC,
21                                     wd.LICZBA_WOLNYCH_MIEJSC)
22     BULK COLLECT INTO result_table
23     FROM VIEW_WYCIECZKI_DOSTEPNE2 wd
24     WHERE wd.KRAJ = fnc_dostepne_wycieczki2.kraj AND data_od <= wd.DATA
25           AND wd.DATA <= data_do;
26
27     RETURN result_table;
28 END fnc_dostepne_wycieczki2;
```

11 Zmienione procedury

11.1 Dodaj rezerwacje 2

```
1  — PROCEDURE DODAJ_REZERWACJE2
2  CREATE OR REPLACE
3  PROCEDURE prc_dodaj_rezerwacje2
4      (id_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE, id_osoby OSOBY.ID_OSOBY%TYPE) AS
5      is_table          INTEGER;
6      new_reservation_id  INTEGER;
7      BEGIN
8          SELECT COUNT(*) INTO is_table
9          FROM OSOBY o
10         WHERE o.ID_OSOBY = prc_dodaj_rezerwacje2.id_osoby;
11
12         IF is_table = 0 THEN
13             raise_application_error(-20014, 'Brak osoby o podanym id');
14         END IF;
15
16         SELECT COUNT(*) INTO is_table
17         FROM VIEW_WYCIECZKI_DOSTEPNE2 wd
18         WHERE wd.ID_WYCIECZKI = prc_dodaj_rezerwacje2.id_wycieczki;
19
20         IF is_table = 0 THEN
21             raise_application_error(-20014, 'Brak dostępnej wycieczki o podanym id
22         ');
23         END IF;
24
25         SELECT COUNT(*) INTO is_table
26         FROM REZERWACJE r
27         WHERE r.ID_WYCIECZKI = prc_dodaj_rezerwacje2.id_wycieczki
28               AND r.ID_OSOBY = prc_dodaj_rezerwacje2.id_osoby;
29
30         IF is_table > 0 THEN
31             raise_application_error(-20014, 'Rezerwacja już istnieje');
32         END IF;
33
34         INSERT INTO REZERWACJE r (r.ID_WYCIECZKI, r.ID_OSOBY, r.STATUS)
35         VALUES (id_wycieczki, id_osoby, 'N')
36         RETURNING r.NR_REZERWACJI INTO new_reservation_id;
37
38         UPDATE WYCIECZKI w
39         SET LICZBA_WOLNYCH_MIEJSC = LICZBA_WOLNYCH_MIEJSC - 1
40         WHERE w.ID_WYCIECZKI = prc_dodaj_rezerwacje2.id_wycieczki;
41
42         INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
43         VALUES (new_reservation_id, CURRENT_DATE, 'N');
44     END prc_dodaj_rezerwacje2;
```

11.2 Zmien status rezerwacji 2

```
1  — PROCEDURE ZMIEN_STATUS_REZERWACJI2
2  CREATE OR REPLACE
3  PROCEDURE prc_zmien_status_rezerwacji2
4      (nr_rezerwacji REZERWACJE.NR_REZERWACJI%TYPE, status REZERWACJE.STATUS%TYPE)
5      AS
6      is_table          INTEGER;
7      past_status        REZERWACJE.STATUS%TYPE;
```



```

7 new_places INTEGER;
8 BEGIN
9     SELECT COUNT(*) INTO is_table
10    FROM REZERWACJE r
11   WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji2.nr_rezerwacji;
12
13     IF is_table = 0 THEN
14         raise_application_error(-20014, 'Brak rezerwacji o podanym nr');
15     END IF;
16
17     SELECT COUNT(*) INTO is_table
18    FROM VIEW_REZERWACJE_W_PRZYSZLOSCI rwp
19   WHERE rwp.NR_REZERWACJI = prc_zmien_status_rezerwacji2.nr_rezerwacji;
20
21     IF is_table = 0 THEN
22         raise_application_error(-20014, 'Status rezerwacji mozna zmieniac
23 tylko ' ||
24                                     'dla wycieczek przyszlosci');
25     END IF;
26
27     SELECT COUNT(*) INTO is_table
28    FROM VIEW_WYCIEZKI_DOSTEPNE2 wd
29   JOIN REZERWACJE r ON r.ID_WYCIEZKI = wd.ID_WYCIEZKI
30  WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji2.nr_rezerwacji;
31
32     SELECT r.STATUS INTO past_status
33    FROM REZERWACJE r
34   WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji2.nr_rezerwacji;
35
36     IF prc_zmien_status_rezerwacji2.past_status = 'A' AND is_table = 0 THEN
37         raise_application_error(-20014, 'Brak dostepnych miejsc');
38     END IF;
39
40     IF prc_zmien_status_rezerwacji2.past_status =
prc_zmien_status_rezerwacji2.status THEN
41         raise_application_error(-20014, 'Nie mozna zamienic status na taki
42 sam');
43     END IF;
44
45     IF prc_zmien_status_rezerwacji2.status = 'A' THEN
46         new_places := 1;
47     ELSE
48         new_places := 0;
49     END IF;
50
51     UPDATE REZERWACJE r
52    SET STATUS = prc_zmien_status_rezerwacji2.status
53   WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji2.nr_rezerwacji;
54
55     UPDATE WYCIEZKI w
56    SET w.LICZBA_WOLNYCH_MIEJSC = w.LICZBA_WOLNYCH_MIEJSC +
prc_zmien_status_rezerwacji2.new_places
57   WHERE w.ID_WYCIEZKI = (SELECT ID_WYCIEZKI FROM REZERWACJE r
58                           WHERE r.NR_REZERWACJI =
prc_zmien_status_rezerwacji2.nr_rezerwacji);
59
60     INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
61    VALUES (prc_zmien_status_rezerwacji2.nr_rezerwacji, CURRENT_DATE,
prc_zmien_status_rezerwacji2.status);
62 END prc_zmien_status_rezerwacji2;

```

11.3 Zmien liczbe miejsc 2

```
1  — PROCEDURE ZMIEN LICZBE MIEJSC2
2  CREATE OR REPLACE
3  PROCEDURE prc_zmien_liczbe_miejsc2
4      (id_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE, liczba_miejsc WYCIECZKI.
      LICZBA_MIEJSC%TYPE) AS
5      is_table      INTEGER;
6      places        INTEGER;
7      free_places   INTEGER;
8      BEGIN
9          SELECT COUNT(*) INTO is_table
10         FROM WYCIECZKI w
11         WHERE w.ID_WYCIECZKI = prc_zmien_liczbe_miejsc2.id_wycieczki;
12
13         IF is_table = 0 THEN
14             raise_application_error(-20014, 'Nie istnieje wycieczki o
podanym id');
15         END IF;
16
17         SELECT COUNT(*) INTO is_table
18         FROM VIEW_WYCIECZKI_W_PRZYSZLOSCI2 wwp
19         WHERE wwp.ID_WYCIECZKI = prc_zmien_liczbe_miejsc2.id_wycieczki;
20
21         IF is_table = 0 THEN
22             raise_application_error(-20014, 'Nie mozna zmieniac liczbe
miejsc skonczonej' ||
23                                     ' wycieczki');
24         END IF;
25
26         SELECT wd.LICZBA_MIEJSC INTO places
27         FROM VIEW_WYCIECZKI_DOSTEPNE2 wd
28         WHERE wd.ID_WYCIECZKI = prc_zmien_liczbe_miejsc2.id_wycieczki;
29
30         IF prc_zmien_liczbe_miejsc2.liczba_miejsc < places THEN
31
32             SELECT wd.LICZBA_WOLNYCH_MIEJSC INTO free_places
33             FROM VIEW_WYCIECZKI_DOSTEPNE2 wd
34             WHERE wd.ID_WYCIECZKI = prc_zmien_liczbe_miejsc2.id_wycieczki;
35
36             IF places - free_places > prc_zmien_liczbe_miejsc2.liczba_miejsc
37 THEN
38                 raise_application_error(-20014, 'Nie mozna zmienic liczbe
miejsc na mniejsza' ||
39                                     ' niz zarejestrowanych
uczestnikow');
40             END IF;
41         END IF;
42
43         UPDATE WYCIECZKI w
44         SET w.LICZBA_MIEJSC = prc_zmien_liczbe_miejsc2.liczba_miejsc ,
45            w.LICZBA_WOLNYCH_MIEJSC = w.LICZBA_WOLNYCH_MIEJSC +
46            (prc_zmien_liczbe_miejsc2.liczba_miejsc - w.LICZBA_MIEJSC)
47         WHERE ID_WYCIECZKI = prc_zmien_liczbe_miejsc2.id_wycieczki;
48     END prc_zmien_liczbe_miejsc2;
```

12 Dodanie triggerów

12.1 Trigger obsługujący dodanie rezerwacji

```
1 CREATE OR REPLACE
2 TRIGGER trigger_dodaj_rezerwacje
3   AFTER INSERT
4   ON REZERWACJE
5   FOR EACH ROW
6 BEGIN
7   INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
8   VALUES (:NEW.NR_REZERWACJI, CURRENT_DATE, :NEW.STATUS);
9
10  UPDATE WYCIECZKI w
11  SET LICZBA_WOLNYCH_MIEJSC = LICZBA_WOLNYCH_MIEJSC - 1
12  WHERE w.ID_WYCIECZKI = :NEW.ID_WYCIECZKI;
13 END;
```

12.2 Trigger obsługujący zmianę statusu

```
1 CREATE OR REPLACE
2 TRIGGER trigger_zmien_status_rezerwacji
3   AFTER UPDATE
4   ON REZERWACJE
5   FOR EACH ROW
6 DECLARE
7   new_places INT;
8 BEGIN
9   INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
10  VALUES (:NEW.NR_REZERWACJI, CURRENT_DATE, :NEW.STATUS);
11
12  IF :NEW.STATUS = 'A' THEN
13    new_places := 1;
14  ELSE
15    new_places := 0;
16  END IF;
17
18  UPDATE WYCIECZKI w
19  SET LICZBA_WOLNYCH_MIEJSC = LICZBA_WOLNYCH_MIEJSC + new_places
20  WHERE w.ID_WYCIECZKI = :NEW.ID_WYCIECZKI;
21 END trigger_zmien_status_rezerwacji;
```

12.3 Trigger zabraniający usunięcia rezerwacji

```
1 CREATE OR REPLACE
2 TRIGGER trigger_zabron_usuwanie_rezerwacji
3   BEFORE DELETE
4   ON REZERWACJE
5   FOR EACH ROW
6 BEGIN
7   raise_application_error(-20014, 'Removing reservation forbidden.');
```

12.4 Trigger obsługujący zmianę liczby miejsc na poziomie wycieczki

```
1 CREATE OR REPLACE
2 TRIGGER trigger_zmien_liczbe_miejsc
3   BEFORE UPDATE OF LICZBA_MIEJSC
4   ON WYCIECZKI
5   FOR EACH ROW
6 BEGIN
7   SELECT :OLD.LICZBA_WOLNYCH_MIEJSC +
8     (:NEW.LICZBA_MIEJSC - :OLD.LICZBA_MIEJSC) INTO :NEW.
9     LICZBA_WOLNYCH_MIEJSC
10  FROM Dual;
11 END;
```

13 Dostosowane procedury

13.1 Dodaj rezerwacje 3

```
1 — PROCEDURE DODAJ_REZERWACJE3
2 CREATE OR REPLACE
3 PROCEDURE prc_dodaj_rezerwacje3
4   (id_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE, id_osoby OSOBY.ID_OSOBY%TYPE) AS
5   is_table          INTEGER;
6   new_reservation_id  INTEGER;
7 BEGIN
8   SELECT COUNT(*) INTO is_table
9   FROM OSOBY o
10  WHERE o.ID_OSOBY = prc_dodaj_rezerwacje3.id_osoby;
11
12  IF is_table = 0 THEN
13    raise_application_error(-20014, 'Brak osoby o podanym id');
14  END IF;
15
16  SELECT COUNT(*) INTO is_table
17  FROM VIEW_WYCIECZKI_DOSTEPNE2 wd
18  WHERE wd.ID_WYCIECZKI = prc_dodaj_rezerwacje3.id_wycieczki;
19
20  IF is_table = 0 THEN
21    raise_application_error(-20014, 'Brak dostępnej wycieczki o podanym id
22  ');
23  END IF;
24
25  SELECT COUNT(*) INTO is_table
26  FROM REZERWACJE r
27  WHERE r.ID_WYCIECZKI = prc_dodaj_rezerwacje3.id_wycieczki
28    AND r.ID_OSOBY = prc_dodaj_rezerwacje3.id_osoby;
29
30  IF is_table > 0 THEN
31    raise_application_error(-20014, 'Rezerwacja już istnieje');
32  END IF;
33
34  INSERT INTO REZERWACJE r (r.ID_WYCIECZKI, r.ID_OSOBY, r.STATUS)
35  VALUES (id_wycieczki, id_osoby, 'N')
36  RETURNING r.NR_REZERWACJI INTO new_reservation_id;
37 END prc_dodaj_rezerwacje3;
```

13.2 Zmien status rezerwacji 3

```
1  — PROCEDURE ZMIEN_STATUS_REZERWACJI3
2  CREATE OR REPLACE
3  PROCEDURE prc_zmien_status_rezerwacji3
4      (nr_rezerwacji REZERWACJE.NR_REZERWACJI%TYPE, status REZERWACJE.STATUS%TYPE)
5      AS
6      is_table      INTEGER;
7      past_status   REZERWACJE.STATUS%TYPE;
8      BEGIN
9          SELECT COUNT(*) INTO is_table
10         FROM REZERWACJE r
11         WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji3.nr_rezerwacji;
12
13         IF is_table = 0 THEN
14             raise_application_error(-20014, 'Brak rezerwacji o podanym nr');
15         END IF;
16
17         SELECT COUNT(*) INTO is_table
18         FROM VIEW_REZERWACJE_W_PRZYSZLOSCI rwp
19         WHERE rwp.NR_REZERWACJI = prc_zmien_status_rezerwacji3.nr_rezerwacji;
20
21         IF is_table = 0 THEN
22             raise_application_error(-20014, 'Status rezerwacji mozna zmieniac
23             tylko ' ||
24                                     'dla wycieczek przyszlosci');
25         END IF;
26
27         SELECT COUNT(*) INTO is_table
28         FROM VIEW_WYCIECZKI_DOSTEPNE2 wd
29         JOIN REZERWACJE r ON r.ID_WYCIECZKI = wd.ID_WYCIECZKI
30         WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji3.nr_rezerwacji;
31
32         SELECT r.STATUS INTO past_status
33         FROM REZERWACJE r
34         WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji3.nr_rezerwacji;
35
36         IF prc_zmien_status_rezerwacji3.past_status = 'A' AND is_table = 0 THEN
37             raise_application_error(-20014, 'Brak dostepnych miejsc');
38         END IF;
39
40         IF prc_zmien_status_rezerwacji3.past_status =
41         prc_zmien_status_rezerwacji3.status THEN
42             raise_application_error(-20014, 'Nie mozna zamienic status na taki
43             sam');
44         END IF;
45
46         UPDATE REZERWACJE r
47         SET STATUS = prc_zmien_status_rezerwacji3.status
48         WHERE r.NR_REZERWACJI = prc_zmien_status_rezerwacji3.nr_rezerwacji;
49     END prc_zmien_status_rezerwacji3;
```

13.3 Zmien liczbe miejsc 3

```
1  — PROCEDURE ZMIEN LICZBE MIEJSC3
2  CREATE OR REPLACE
3  PROCEDURE prc_zmien_liczbe_miejsc3
4      (id_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE, liczba_miejsc WYCIECZKI.
      LICZBA_MIEJSC%TYPE) AS
5      is_table      INTEGER;
6      places        INTEGER;
7      free_places   INTEGER;
8      BEGIN
9          SELECT COUNT(*) INTO is_table
10         FROM WYCIECZKI w
11         WHERE w.ID_WYCIECZKI = prc_zmien_liczbe_miejsc3.id_wycieczki;
12
13         IF is_table = 0 THEN
14             raise_application_error(-20014, 'Nie istnieje wycieczki o
podanym id');
15         END IF;
16
17         SELECT COUNT(*) INTO is_table
18         FROM VIEW_WYCIECZKI_W_PRZYSZLOSCI2 wwp
19         WHERE wwp.ID_WYCIECZKI = prc_zmien_liczbe_miejsc3.id_wycieczki;
20
21         IF is_table = 0 THEN
22             raise_application_error(-20014, 'Nie mozna zmieniac liczbe
miejsc skonczonej' ||
23                                     ' wycieczki');
24         END IF;
25
26         SELECT wd.LICZBA_MIEJSC INTO places
27         FROM VIEW_WYCIECZKI_DOSTEPNE2 wd
28         WHERE wd.ID_WYCIECZKI = prc_zmien_liczbe_miejsc3.id_wycieczki;
29
30         IF prc_zmien_liczbe_miejsc3.liczba_miejsc < places THEN
31
32             SELECT wd.LICZBA_WOLNYCH_MIEJSC INTO free_places
33             FROM VIEW_WYCIECZKI_DOSTEPNE2 wd
34             WHERE wd.ID_WYCIECZKI = prc_zmien_liczbe_miejsc3.id_wycieczki;
35
36             IF places - free_places > prc_zmien_liczbe_miejsc3.liczba_miejsc
THEN
37                 raise_application_error(-20014, 'Nie mozna zmienic liczbe
miejsc na mniejsza' ||
38                                     ' niz zarejestrowanych
uczestnikow');
39             END IF;
40         END IF;
41
42         UPDATE WYCIECZKI
43         SET LICZBA_MIEJSC = prc_zmien_liczbe_miejsc3.liczba_miejsc
44         WHERE ID_WYCIECZKI = prc_zmien_liczbe_miejsc3.id_wycieczki;
45     END prc_zmien_liczbe_miejsc3;
```