

# Laboratorium 1

## PL/SQL

AGATA NOWARA

<b>TABELE:</b>	<b>3</b>
1.1 TABELA OSOBY	3
1.2 TABELA WYCIECZKI	3
1.3 TABELA REZERWACJE	4
1.4 CONSTRAINTY	4
<b>2.WYPEŁNIENIE TABEL DANymi</b>	<b>5</b>
2.1 WYPEŁNIONA TABELA OSOBY	7
2.2 WYPEŁNIONA TABELA WYCIECZKI	7
2.3 WYPEŁNIONA TABELA REZERWACJE	7
<b>3. WIDOKI</b>	<b>7</b>
3.1 WSZYSTKIE REZERWACJE	7
3.2 REZERWACJE POTWIERDZONE	8
3.3 REZERWACJE W PRZYSZŁOŚCI	8
3.4 LICZBA WOLNYCH MIEJSC NA WYCIECZCE	9
3.5 DOSTĘPNE WYCIECZKI	9
<b>4. PROCEDURY/FUNKCJE POBIERAJĄCE DANE</b>	<b>10</b>
4.1 UCZESTNICY WYCIECZKI	10
4.2 REZERWACJE OSOBY	11
4.3 DOSTĘPNE WYCIECZKI	12
<b>5. PROCEDURY MODYFIKUJĄCE DANE</b>	<b>14</b>
5.1 DODAJ REZERWACJĘ	14
5.2 ZMIENŃ STATUS REZERWACJI	16
5.3 ZMIENŃ LICZBĘ MIEJSC	19
<b>6. TABELA DZIENNIKUJĄCA ZMIANY STATUSU</b>	<b>21</b>
<b>7. ZMIANA STRUKTURY BAZY DANYCH</b>	<b>21</b>
7.1 DODANIE ATRYBUTU LICZBA_WOLNYCH_MIEJSC	21
7.2 PROCEDURA PRZELICZ	21
7.3 MODYFIKACJA WIDOKÓW	22
7.3.1 WIDOK WYCIECZKI MIEJSCA 2	22

7.4 MODYFIKACJA PROCEDUR POBIERAJĄCYCH	22
7.4.1 PROCEDURA DOSTĘPNE MIEJSCA 2	22
7.5 MODYFIKACJA PROCEDUR MODYFIKUJĄCYCH	23
7.5.1 PROCEDURA DODAJ REZERWACJE 2	23
7.5.2 PROCEDURA ZMIEN STATUS REZERWACJI 2	25
7.5.3 PROCEDURA ZMIEN LICZBĘ MIEJSC 2	27
<b>8. DODANIE TRIGGERÓW</b>	<b>28</b>
8.1 TRIGGER DODAJ REZERWACJE	28
8.2 TRIGGER ZMIANA STATUSU REZERWACJI	29
8.3 TRIGGER ZMIANA LICZBY MIEJSC	29
8.4 TRIGGER ZAKAZ USUWANIA REZERWACJI	29
8.5 DRUGA MODYFIKACJA PROCEDURY DODAJ REZERWACJE	30
8.6 DRUGA MODYFIKACJA PROCEDURY ZMIEN STATUS REZERWACJI	31
8.7 DRUGA MODYFIKACJA PROCEDURY ZMIEN LICZBĘ MIEJSC	33

# 1.TABELE:

## 1.1 TABELA OSOBY

```
CREATE TABLE OSOBY
(
    ID_OSOBY INT GENERATED ALWAYS AS IDENTITY NOT NULL
    , IMIE VARCHAR2(50)
    , NAZWISKO VARCHAR2(50)
    , PESEL VARCHAR2(11)
    , KONTAKT VARCHAR2(100)
    , CONSTRAINT OSOBY_PK PRIMARY KEY
      (
        ID_OSOBY
      )
    ENABLE
);
```

## 1.2 TABELA WYCIECZKI

```
CREATE TABLE WYCIECZKI
(
    ID_WYCIECZKI INT GENERATED ALWAYS AS IDENTITY NOT NULL
    , NAZWA VARCHAR2(100)
    , KRAJ VARCHAR2(50)
    , DATA DATE
    , OPIS VARCHAR2(200)
    , LICZBA_MIEJSC INT
    , CONSTRAINT WYCIECZKI_PK PRIMARY KEY
      (
        ID_WYCIECZKI
      )
    ENABLE
);
```

## 1.3 TABELA REZERWACJE

```
CREATE TABLE REZERWACJE
(
    NR_REZERWACJI INT GENERATED ALWAYS AS IDENTITY NOT NULL
    , ID_WYCIECZKI INT
    , ID_OSOBY INT
    , STATUS CHAR(1)
    , CONSTRAINT REZERWACJE_PK PRIMARY KEY
    (
        NR_REZERWACJI
    )
    ENABLE
);
```

## 1.4 CONSTRAINTY

```
ALTER TABLE REZERWACJE
ADD CONSTRAINT REZERWACJE_FK1 FOREIGN KEY
(
    ID_OSOBY
)
REFERENCES OSOBY
(
    ID_OSOBY
)
ENABLE;
```

```
ALTER TABLE REZERWACJE
ADD CONSTRAINT REZERWACJE_FK2 FOREIGN KEY
(
    ID_WYCIECZKI
)
REFERENCES WYCIECZKI
(
    ID_WYCIECZKI
)
ENABLE;
```

```
ALTER TABLE REZERWACJE
ADD CONSTRAINT REZERWACJE_CHK1 CHECK
(status IN ('N', 'P', 'Z', 'A'))
ENABLE;
```

## 2.WYPEŁNIENIE TABEL DANYMI

```
--TABELA OSOBY
```

```
--1
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Adam', 'Kowalski', '87654321', 'tel: 6623');
```

```
--2
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Jan', 'Nowak', '12345678', 'tel: 2312, dzwonić po 18.00');
```

```
--3
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Aleksandra', 'Noga', '76052309865', 'tel:3456');
```

```
--4
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Marek', 'Kaczka', '98012700456', 'tel: 7723');
```

```
--5
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Karolina', 'Lis', '89041000809', 'tel:6606');
```

```
--6
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Tomasz', 'Cebula', '77031901204', 'tel:2245');
```

```
--7
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Alicja', 'Jura', '99070700801', 'tel:5601');
```

```
--8
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Tadeusz', 'Bywalec', '88012804502', 'tel:5506');
```

```
--9
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Jakub', 'Lysy', '71020200606', 'tel:7286');
```

```
--10
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Helena', 'Szczodra', '66063000803', 'tel:6776');
```

```
--TABELA WYCIEZKI
```

```
--1
```

```
INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba_miejsc)
VALUES ('Wycieczka do Paryża', 'Francja', TO_DATE('2016-01-01',
'YYYY-MM-DD'), 'Ciekawa wycieczka ...', 3);
```

```
--2
```

```
INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba_miejsc)
VALUES ('Piękny Kraków', 'Polska', TO_DATE('2017-02-03', 'YYYY-MM-DD'),
```

```

'Najciekawsza wycieczka ...', 2 );
--3
INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba_miejsc)
VALUES ('Wieliczka', 'Polska', TO_DATE('2017-03-03', 'YYYY-MM-DD'),
'Zadziwiająca kopalnia ...', 2);
--4
INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba_miejsc)
VALUES ('Cesarskie Chiny', 'Chiny', TO_DATE('2020-06-02', 'YYYY-MM-DD'),
'Niesamowita podróż po ...', 5);

--TABELA REZERWACJE
--1
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (1,1,'N');
--2
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (2,2,'P');
--3
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (3,3,'Z');
--4
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (4,4,'A');
--5
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (1,5,'A');
--6
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (2,6,'Z');
--7
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (3,7,'P');
--8
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (4,8,'N');
--9
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES (4,9,'P');
--10
INSERT INTO rezerwacje(id_wycieczki, id_osoby, status) VALUES
(2,11,'A');

```

## 2.1 WYPEŁNIONA TABELA OSOBY

	ID_OSOBY	IMIE	NAZWISKO	PESEL	KONTAKT
1	1	Adam	Kowalski	87654321	tel: 6623
2	2	Jan	Nowak	12345678	tel: 2312, dzwonić po 18.00
3	3	Aleksandra	Noga	76052309865	tel:3456
4	4	Marek	Kaczka	98012700456	tel: 7723
5	5	Karolina	Lis	89041000809	tel:6606
6	6	Tomasz	Cebula	77031901204	tel:2245
7	7	Alicja	Jura	99070700801	teel:5601
8	8	Tadeusz	Bywalec	88012804502	tel:5506
9	9	Jakub	Lysy	71020200606	tel:7286
10	11	Helena	Szczodra	66063000803	tel:6776

## 2.2 WYPEŁNIONA TABELA WYCIECZKI

	ID_WYCIECZKI	NAZWA	KRAJ	DATA	OPIS	LICZBA_MIEJSC
1	1	Wycieczka do Paryża	Francja	2016-01-01 00:00:00	Ciekawa wycieczka ...	3
2	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Najciekawsza wycieczka ...	2
3	3	Wieliczka	Polska	2017-03-03 00:00:00	Zadziwiająca kopalnia ...	2
4	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Niesamowita podróż po ...	5

## 2.3 WYPEŁNIONA TABELA REZERWACJE

	NR_REZERWACJI	ID_WYCIECZKI	ID_OSOBY	STATUS
1	1	1	1	N
2	2	2	2	P
3	3	3	3	Z
4	4	4	4	A
5	5	1	5	A
6	6	2	6	Z
7	7	3	7	P
8	8	4	8	N
9	9	4	9	P
10	10	2	11	A

## 3. WIDOKI

### 3.1 WSZYSTKIE REZERWACJE

```
CREATE VIEW RezerwacjeWszystkie
AS
SELECT w.ID_WYCIECZKI, w.NAZWA, w.KRAJ, w.DATA, o.IMIE,
o.NAZWISKO, r.STATUS
FROM WYCIECZKI w
JOIN REZERWACJE r ON w.ID_WYCIECZKI = r.ID_WYCIECZKI
JOIN OSOBY o ON r.ID_OSOBY = o.ID_OSOBY;
```

	ID_WYCIECZKI	NAZWA	KRAJ	DATA	IMIE	NAZWISKO	STATUS
1	1	Wycieczka do Paryża	Francja	2016-01-01 00:00:00	Adam	Kowalski	N
2	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Jan	Nowak	P
3	3	Wieliczka	Polska	2017-03-03 00:00:00	Aleksandra	Noga	Z
4	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Marek	Kaczka	A
5	1	Wycieczka do Paryża	Francja	2016-01-01 00:00:00	Karolina	Lis	A
6	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Tomasz	Cebula	Z
7	3	Wieliczka	Polska	2017-03-03 00:00:00	Alicja	Jura	P
8	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Tadeusz	Bywalec	N
9	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Jakub	Lysy	P
10	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Helena	Szczodra	A

## 3.2 REZERWACJE POTWIERDZONE

```
CREATE VIEW RezerwacjePotwierdzone
AS
SELECT *
FROM RezerwacjeWszystkie
WHERE status='P' or status='Z';
```

	ID_WYCIECZKI	NAZWA	KRAJ	DATA	IMIE	NAZWISKO	STATUS
1	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Jan	Nowak	P
2	3	Wieliczka	Polska	2017-03-03 00:00:00	Aleksandra	Noga	Z
3	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Tomasz	Cebula	Z
4	3	Wieliczka	Polska	2017-03-03 00:00:00	Alicja	Jura	P
5	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Jakub	Lysy	P

## 3.3 REZERWACJE W PRZYSZŁOŚCI

```
CREATE VIEW RezerwacjeWPrzyszlosci
AS
SELECT *
FROM RezerwacjeWszystkie
WHERE data>sysdate;
```

	ID_WYCIECZKI	NAZWA	KRAJ	DATA	IMIE	NAZWISKO	STATUS
1	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Marek	Kaczka	A
2	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Tadeusz	Bywalec	N
3	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Jakub	Lysy	P

## 3.4 LICZBA WOLNYCH MIEJSC NA WYCIECZCE

```
CREATE VIEW WycieczkiMiejsca
AS
SELECT w.ID_WYCIECZKI, w.KRAJ, w.DATA, w.NAZWA, w.LICZBA_MIEJSC,
(w.LICZBA_MIEJSC- COUNT(DISTINCT r.ID_OSOBY))as LICZBA_WOLNYCH_MIEJSC
FROM WYCIECZKI w
LEFT JOIN (SELECT * FROM REZERWACJE r
WHERE r.status <> 'A') r
ON r.ID_WYCIECZKI=w.ID_WYCIECZKI
GROUP BY w.ID_WYCIECZKI, w.KRAJ, w.DATA, w.NAZWA,
w.LICZBA_MIEJSC;
```



	ID_WYCIECZKI	KRAJ	DATA	NAZWA	LICZBA_MIEJSC	LICZBA_WOLNYCH_MIEJSC
1		1 Francja	2016-01-01 00:00:00	Wycieczka do Paryża	3	2
2		2 Polska	2017-02-03 00:00:00	Piękny Kraków	2	0
3		3 Polska	2017-03-03 00:00:00	Wieliczka	2	0
4		4 Chiny	2020-06-02 00:00:00	Cesarskie Chiny	5	3

## 3.5 DOSTĘPNE WYCIECZKI

```
CREATE VIEW WycieczkiDostepne
AS
SELECT *
FROM WycieczkiMiejsca
WHERE LICZBA_WOLNYCH_MIEJSC > 0 and DATA > sysdate;
```

	ID_WYCIECZKI	KRAJ	DATA	NAZWA	LICZBA_MIEJSC	LICZBA_WOLNYCH_MIEJSC
1		4 Chiny	2020-06-02 00:00:00	Cesarskie Chiny	5	3

## 4. PROCEDURY/FUNKCJE POBIERAJĄCE DANE

### 4.1 UCZESTNICY WYCIECZKI

TWORZENIE OBIEKTU:

```
CREATE OR REPLACE TYPE UCZESTNIK_WYCIECZKI AS OBJECT
(
    NAZWA      NVARCHAR2(100),
    KRAJ       NVARCHAR2(50),
    DATA      DATE,
    IMIE       NVARCHAR2(50),
    NAZWISKO   NVARCHAR2(50),
    STATUS     CHAR(1)
);
```

```
CREATE OR REPLACE TYPE UCZESTNICY_WYCIECZKI_TABLE IS TABLE OF
UCZESTNIK_WYCIECZKI;
```

FUNKCJA:

```
CREATE OR REPLACE FUNCTION UCZESTNICZY_WYCIECZKI
(
    id WYCIECZKI.ID_WYCIECZKI%TYPE
)
RETURN UCZESTNICZY_WYCIECZKI_TABLE AS
    result UCZESTNICZY_WYCIECZKI_TABLE;
trip_exists INT;
BEGIN
    -- check if trip exists; trip_exists==0 => trip with given ID does
    not exist
    SELECT COUNT(*) INTO trip_exists
    FROM WYCIECZKI
    WHERE WYCIECZKI.ID_WYCIECZKI = UCZESTNICZY_WYCIECZKI.id;
    IF trip_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID does not
    exist.');
```

```
    END IF;

    SELECT UCZESTNIK_WYCIECZKI(w.NAZWA, w.KRAJ, w.DATA, o.IMIE,
    o.NAZWISKO, r.STATUS)
    -- bulk added to make this operation more efficient
    BULK COLLECT INTO result
    FROM WYCIECZKI w
        JOIN REZERWACJE r
            ON w.ID_WYCIECZKI = r.ID_WYCIECZKI
        JOIN OSOBY o
            ON r.ID_OSOBY = o.ID_OSOBY
    WHERE w.ID_WYCIECZKI = id AND r.STATUS <> 'A';
    RETURN result;
END;
```

SPRAWDZENIE POPRAWNOŚCI DZIAŁANIA:

Otrzymany rezultat jest zgodny z oczekiwanym.

```
SELECT UCZESTNICZY_WYCIECZKI(3) from dual;
```

```
"UCZESTNICZY_WYCIECZKI(3)"
1  {{Wieliczka,Polska,2017-03-03 00:00:00,Aleksandra,Noga,Z},{Wieliczka,Polska,2017-03-03 00:00:00,Alicja,Jura,P}}

```

## 4.2 REZERWACJE OSOBY

TWORZENIE OBIEKTU:

```
CREATE OR REPLACE TYPE REZERWACJA_OSOBY AS OBJECT
(
    NAZWA                VARCHAR2(100),
    KRAJ                 VARCHAR2(50),
    DATA                DATE,
    STATUS               CHAR(1)
);
```

```
CREATE OR REPLACE TYPE REZERWACJE_OSOBY_TABLE IS TABLE OF
REZERWACJA_OSOBY;
```

FUNKCJA:

```
CREATE OR REPLACE FUNCTION REZERWACJE_OSOBY
(
    id OSOBY.ID_OSOBY%TYPE
)
RETURN REZERWACJE_OSOBY_TABLE AS
    result REZERWACJE_OSOBY_TABLE;
reservation_exists INT;
BEGIN
    -- check if person with given ID has any reservation;
    reservation_exists > 0 => this person has reservation(s)
    SELECT COUNT(*) INTO reservation_exists
    FROM REZERWACJE r
    WHERE r.ID_OSOBY = id;
    IF reservation_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-2000, 'Person with given ID have no
reservations' );
    END IF;
    SELECT REZERWACJA_OSOBY(w.NAZWA, w.KRAJ, w.DATA, r.STATUS)
    -- bulk added to make this operation more efficient
    BULK COLLECT INTO result
    FROM WYCIECZKI w
        JOIN REZERWACJE r
            ON w.ID_WYCIECZKI = r.ID_WYCIECZKI
        JOIN OSOBY o
            ON r.ID_OSOBY = o.ID_OSOBY
    WHERE o.ID_OSOBY = id AND r.STATUS <> 'A';
RETURN result;
```

```
END;
```

SPRAWDZENIE POPRAWNOŚCI DZIAŁANIA:

Otrzymany rezultat jest zgodny z oczekiwanym.

```
SELECT REZERWACJE_OSOBY(6) from dual
```

	"REZERWACJE_OSOBY(6)"
1	{{Piękny Kraków,Polska,2017-02-03 00:00:00,Z}}

## 4.3 DOSTĘPNE WYCIECZKI

TWORZENIE OBIEKTU:

```
CREATE OR REPLACE TYPE WYCIECZKA AS OBJECT
(
  ID_wycieczki          NUMBER,
  NAZWA                 VARCHAR2(100),
  KRAJ                 VARCHAR2(50),
  DATA                DATE,
  OPIS                 VARCHAR2(200),
  LICZBA_MIEJSC        NUMBER
);
```

```
CREATE OR REPLACE TYPE WYCIECZKI_TABLE IS TABLE OF WYCIECZKA;
```

FUNKCJA:

```
CREATE OR REPLACE FUNCTION DOSTEPNE_WYCIECZKI
(
  kraj WYCIECZKI.KRAJ%TYPE, data_od DATE, data_do DATE
)
RETURN WYCIECZKI_TABLE AS
  result WYCIECZKI_TABLE;
BEGIN
  -- check if data parameters are in correct order
  IF data_od > data_do THEN
    RAISE_APPLICATION_ERROR(-20001, 'Data_od must be before
data_do');
  END IF;
  SELECT WYCIECZKA(w.ID_WYCIECZKI, w.NAZWA, w.KRAJ, w.DATA, w.OPIS,
w.LICZBA_MIEJSC)
```

```
-- bulk added to make this operation more efficient
BULK COLLECT INTO result
FROM WYCIECZKI w
JOIN WycieczkiDostepne wd ON w.ID_WYCIECZKI=wd.ID_WYCIECZKI
WHERE wd.kraj=DOSTEPNE_WYCIECZKI.kraj AND wd.DATA >= data_od AND
wd.DATA <= data_do;
RETURN result;
END;
```

## SPRAWDZANIE POPRAWNOŚCI:

Otrzymany rezultat jest zgodny z oczekiwanym.

```
SELECT DOSTEPNE_WYCIECZKI('Chiny', TO_DATE('2020/05/30'),
TO_DATE('2020/06/30') ) from dual
```

	DOSTEPNE_WYCIECZKI('CHINY',TO_DATE('2020/05/30'),TO_DATE('2020/06/30'))
1	{{4,Cesarskie Chiny,Chiny,2020-06-02 00:00:00,Niesamowita podróż po ...,5}}

## 5. PROCEDURY MODYFIKUJĄCE DANE

### 5.1 DODAJ REZERWACJĘ

W poniżej przedstawionych procedurach uwzględniłam już użycie tablicy REZERWACJE\_LOG, której tworzenie zostało opisane dopiero w punkcie 6.

```
CREATE OR REPLACE PROCEDURE DODAJ_REZERWACJE
(
    ID_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE,
    ID_osoby OSOBY.ID_OSOBY%TYPE
)
AS
    person_exists          INT;
    trip_available          INT;
    reservation_exists     INT;
    new_reservation_ID     INT;
BEGIN
    -- check if person with given ID exists; person_exists==0 => this
    person does not exist
    SELECT COUNT(*)
    INTO person_exists
    FROM OSOBY
```

```

WHERE OSOBY.ID_OSOBY = DODAJ_REZERWACJE.ID_osoby;
IF person_exists = 0 THEN
    RAISE_APPLICATION_ERROR(-20000, 'Person with chosen ID does not
exist.');
```

*END IF;*

*-- check if trip exists and is available;*

*counter trip\_available == 0 => there is no trip available with given ID*

```

SELECT COUNT(*)
INTO trip_available
FROM WycieczkiDostepne wd
WHERE wd.ID_WYCIECZKI = DODAJ_REZERWACJE.ID_wycieczki;

IF trip_available = 0 THEN
    RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID is not
available or does not exist.');
```

*END IF;*

*-- checks if reservation has already existed; counter*

*reservation\_exists==0 => there is no reservation with given trip ID and*

*person ID*

```

SELECT COUNT(*)
INTO reservation_exists
FROM REZERWACJE r
WHERE r.ID_WYCIECZKI = DODAJ_REZERWACJE.ID_wycieczki AND r.ID_OSOBY =
DODAJ_REZERWACJE.ID_osoby;

IF reservation_exists > 0 THEN
    RAISE_APPLICATION_ERROR(-20004, 'Reservation for chosen trip ID
and person ID has already existed.');
```

*END IF;*

```

INSERT INTO REZERWACJE (ID_WYCIECZKI, ID_OSOBY, STATUS)
VALUES (DODAJ_REZERWACJE.ID_wycieczki, DODAJ_REZERWACJE.ID_osoby,
'N')
```

*--save automatically generating NR\_REZERWACJI to make use of it in*

*REZERWACJE\_LOG table*

```

RETURNING NR_REZERWACJI INTO new_reservation_ID;

-- added after creating table REZERWACJE_LOG;
INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
VALUES (new_reservation_ID, CURRENT_DATE, 'N');
```

*--procedures which insert or update sth don't commit automatically*

*COMMIT;*

*END;*

SPRAWDZANIE POPRAWNOŚCI DZIAŁANIA:

## MOŻLIWE BŁĘDY:

BEGIN

DODAJ\_REZERWACJE(6,1);

END;

[72000][20000] ORA-20000: Trip with chosen ID is not available or does not exist.  
ORA-06512: przy "BD\_NOWARA.DODAJ\_REZERWACJE", linia 28  
ORA-06512: przy linia 2  
Position: 0

BEGIN

DODAJ\_REZERWACJE(4,9);

END;

[72000][20004] ORA-20004: Reservation for chosen trip ID and person ID has already existed.  
ORA-06512: przy "BD\_NOWARA.DODAJ\_REZERWACJE", linia 38  
ORA-06512: przy linia 2  
Position: 0

BEGIN

DODAJ\_REZERWACJE(4,20);

END;

[72000][20000] ORA-20000: Person with chosen ID does not exist.  
ORA-06512: przy "BD\_NOWARA.DODAJ\_REZERWACJE", linia 18  
ORA-06512: przy linia 2  
Position: 0

## POPRAWNE DODANIE REZERWACJI

BEGIN

DODAJ\_REZERWACJE(4,11);

END;

## Rezultat dodania rezerwacji do tablicy REZERWACJE

	NR_REZERWACJI	ID_WYCIEZKI	ID_OSOBY	STATUS
1	1	1	1	N
2	2	2	2	P
3	3	3	3	Z
4	4	4	4	A
5	5	1	5	A
6	6	2	6	Z
7	7	3	7	P
8	8	4	8	N
9	9	4	9	P
10	10	2	11	A
11	21	4	11	N

## Rezultat dodania rezerwacji do tablicy dziennikującej REZERWACJE\_LOG

	ID	ID_REZERWACJI	DATA	STATUS
1	1	21	2020-03-07 18:19:08	N

## 5.2 ZMIENÍ STATUS REZERWACJI

```
CREATE PROCEDURE ZMIEN_STATUS_REZERWACJI
(
    ID_rezerwacji REZERWACJE.NR_REZERWACJI%TYPE,
    status REZERWACJE.STATUS%TYPE
)
AS
    not_updated_status REZERWACJE.STATUS%TYPE;
    reservation_exists INT;
BEGIN
    -- check if reservation exists (in future); counter
    reservation_exists==0 => there is no reservation with given ID
    SELECT COUNT(*)
    INTO reservation_exists
    FROM RezerwacjeWPrzyszlosci rwp
    JOIN REZERWACJE r ON r.ID_WYCIECZKI=rwp.ID_WYCIECZKI
    JOIN OSOBY o ON r.ID_OSOBY = o.ID_OSOBY
    WHERE r.NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI.ID_rezerwacji;

    IF reservation_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'There is no reservation with
given ID');
    END IF;

    --showing different possibilities and errors of changing status
    depending on not_updated_status value
    SELECT STATUS
    INTO not_updated_status
    FROM REZERWACJE r
    WHERE r.NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI.ID_REZERWACJI;
    CASE
        WHEN not_updated_status IS NULL
            THEN RAISE_APPLICATION_ERROR(-20002, 'Reservation with chosen
ID does not exist.');
```

```
        WHEN not_updated_status = 'A'
            THEN RAISE_APPLICATION_ERROR(-20003, 'Status of cancelled
reservation ("A") can not be changed.');
```

```
        WHEN not_updated_status = 'P'
            THEN
                IF ZMIEN_STATUS_REZERWACJI.status NOT IN ('Z','A') THEN
```



```

        RAISE_APPLICATION_ERROR(-20003,
                                'Status of confirmed reservation
("P") can be changed only to "Z" (confirmed and paid) or "A"
(cancelled).');
    END IF;

    WHEN not_updated_status = 'Z'
    THEN
        IF ZMIEN_STATUS_REZERWACJI.status <> 'A' THEN
            RAISE_APPLICATION_ERROR(-20003,
                                    'Status of confirmed and paid
reservation ("Z") can be changed only to "A" (cancelled).');
        END IF;

    ELSE
        RAISE_APPLICATION_ERROR(-30000, 'Internal application error,
CASE not found. ');
    END CASE;

    UPDATE REZERWACJE
    SET STATUS = ZMIEN_STATUS_REZERWACJI.status
    WHERE NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI.ID_rezerwacji;

--added after creating table REZERWACJE_LOG
    INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
    VALUES (ZMIEN_STATUS_REZERWACJI.ID_rezerwacji, CURRENT_DATE,
ZMIEN_STATUS_REZERWACJI.status);

--procedures which insert or update sth don't commit automatically
COMMIT;
END;

```

## SPRAWDZANIE POPRAWNOŚCI DZIAŁANIA:

### MOŻLIWE BŁĘDY

```

BEGIN
    ZMIEN_STATUS_REZERWACJI(20, 'P');
END;

```

```

[72000][20002] ORA-20002: There is no reservation with given ID
ORA-06512: przy "BD_NOWARA.ZMIEN_STATUS_REZERWACJI", linia 17
ORA-06512: przy linia 2
Position: 0

```

```

BEGIN
    ZMIEN_STATUS_REZERWACJI(4, 'P');
END;

```

[72000][20003] ORA-20003: Status of cancelled reservation ("A") can not be changed.  
 ORA-06512: przy "BD\_NOWARA.ZMIEN\_STATUS\_REZERWACJI", linia 29  
 ORA-06512: przy linia 2  
 Position: 0

```
BEGIN
  ZMIEN_STATUS_REZERWACJI(9, 'N');
END;
```

[72000][20003] ORA-20003: Status of confirmed reservation ("P") can be changed only to "Z" (confirmed and paid) or "A" (cancelled).  
 ORA-06512: przy "BD\_NOWARA.ZMIEN\_STATUS\_REZERWACJI", linia 36  
 ORA-06512: przy linia 2  
 Position: 0

## POPRAWNE ZMIENIENIE STATUSU

```
BEGIN
  ZMIEN_STATUS_REZERWACJI(9, 'Z');
END;
```

Rezultat zmienienia statusu rezerwacji widoczny w tablicy REZERWACJE. Zmiana statusu z P na Z.

	NR_REZERWACJI	ID_WYCIEZKI	ID_OSOBY	STATUS
1	1	1	1	N
2	2	2	2	P
3	3	3	3	Z
4	4	4	4	A
5	5	1	5	A
6	6	2	6	Z
7	7	3	7	P
8	8	4	8	N
9	9	4	9	Z
10	10	2	11	A
11	21	4	11	N

Rezultat zmieniania statusu rezerwacji widoczny w tablicy dziennikującej RZERWACJE\_LOG

	ID	ID_REZERWACJI	DATA	STATUS
1	1	21	2020-03-07 18:19:08	N
2	2	9	2020-03-07 18:36:33	Z

## 5.3 ZMIEN LICZBĘ MIEJSC

```

CREATE OR REPLACE PROCEDURE ZMIEN_LICZBE_MIEJSC
(
    ID_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE,
    nowa_liczba_miejsc WYCIECZKI.LICZBA_MIEJSC%TYPE
)
AS
    reserved_places INT;
BEGIN
    SELECT wm.LICZBA_MIEJSC - wm.LICZBA_WOLNYCH_MIEJSC
    INTO reserved_places
    FROM WycieczkiMiejscsca wm
    WHERE wm.ID_WYCIECZKI = ZMIEN_LICZBE_MIEJSC.ID_wycieczki;

    IF reserved_places > nowa_liczba_miejsc OR nowa_liczba_miejsc < 0
    THEN
        RAISE_APPLICATION_ERROR(-20004, 'Given number of places is
inappropriate (smaller than number of reserved places).');
    END IF;

    UPDATE WYCIECZKI
    SET LICZBA_MIEJSC = ZMIEN_LICZBE_MIEJSC.nowa_liczba_miejsc
    WHERE ID_WYCIECZKI = ZMIEN_LICZBE_MIEJSC.ID_wycieczki;

    --procedures which insert or update sth don't commit automatically
    COMMIT;

    --catch exception from SELECT INTO in case of no_data
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID not
found. ');
    END;

```

## SPRAWDZENIE POPRAWNOŚCI DZIAŁANIA:

### MOŻLIWE BŁĘDY:

```

BEGIN
    ZMIEN_LICZBE_MIEJSC(3,1);
END;

```

```

[72000][20004] ORA-20004: Given number of places is inappropriate (smaller than number of reserved places).
ORA-06512: przy "BD.NOWARA.ZMIEN_LICZBE_MIEJSC", linia 16
ORA-06512: przy linia 2
Position: 0

```

```

BEGIN
    ZMIEN_LICZBE_MIEJSC(20, 3);

```

```
END;
```

```
[72000][20000] ORA-20000: Trip with chosen ID not found.  
ORA-06512: przy "BD_NOWARA.ZMIEN_LICZBE_MIEJSC", linia 24  
ORA-06512: przy linia 2  
Position: 0
```

## POPRAWNE ZMIENIENIE ILOŚCI MIEJSC.

Liczba miejsc na wycieczkę Piękny Kraków (o indeksie 2) zmieniona z 2 na 3.

```
BEGIN
```

```
    ZMIEN_LICZBE_MIEJSC(2,3);
```

```
END;
```

	ID_WYCIEZKI	NAZWA	KRAJ	DATA	OPIS	LICZBA_MIEJSC
1	1	Wycieczka do Paryza	Francja	2016-01-01 00:00:00	Ciekawa wycieczka ...	3
2	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Najciekawsza wycieczka ...	3
3	3	Wieliczka	Polska	2017-03-03 00:00:00	Zadziwiająca kopalnia ...	2
4	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Niesamowita podróż po ...	5

## 6. TABELA DZIENNIKUJĄCA ZMIANY STATUSU

```
CREATE TABLE REZERWACJE_LOG  
(  
    ID                INT GENERATED ALWAYS AS IDENTITY NOT NULL,  
    ID_REZERWACJI     INT,  
    DATA              DATE,  
    STATUS             CHAR(1),  
    CONSTRAINT REZERWACJE_LOG_PK PRIMARY KEY (ID)  
ENABLE  
)
```

## 7. ZMIANA STRUKTURY BAZY DANYCH

### 7.1 DODANIE ATRYBUTU LICZBA\_WOLNYCH\_MIEJSC

```
ALTER TABLE WYCIEZKI  
ADD LICZBA_WOLNYCH_MIEJSC INT;
```

### 7.2 PROCEDURA PRZELICZ

```
CREATE OR REPLACE PROCEDURE PRZELICZ AS  
BEGIN  
    UPDATE WYCIEZKI w  
    SET LICZBA_WOLNYCH_MIEJSC=LICZBA_MIEJSC-
```

```

        (SELECT COUNT(*)
        FROM REZERWACJE r
        WHERE r.ID_WYCIECZKI= w.ID_WYCIECZKI
        AND r.STATUS <> 'A');

END;

```

SPRAWDZANIE POPRAWNOŚCI DZIAŁANIA:  
Otrzymane rezultaty zgadzają się ze stanem faktycznym.

```

BEGIN
    PRZELICZ;
END;

```

ID_WYCIECZKI	NAZWA	KRAJ	DATA	OPIS	LICZBA_MIEJSC	LICZBA_WOLNYCH_MIEJSC
1	1 Wycieczka do Paryża	Francja	2016-01-01 00:00:00	Ciekawa wycieczka ...	3	2
2	2 Piękny Kraków	Polska	2017-02-03 00:00:00	Najciekawsza wycieczka ...	3	1
3	3 Wieliczka	Polska	2017-03-03 00:00:00	Zadziwiająca kopalnia ...	2	0
4	4 Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Niesamowita podróż po ...	5	2

## 7.3 MODYFIKACJA WIDOKÓW

Poniżej umieszczono jedyny widok, w którym dokonano korekt ze względu na dodaną powyżej dodatkową kolumnę w tabeli WYCIECZKI. Zmiany polegały na uproszczeniu składni, dzięki użyciu nowo stworzonego atrybutu LICZBA\_WOLNYCH\_MIEJSC

### 7.3.1 WIDOK WYCIECZKI MIEJSCA 2

```

CREATE VIEW WycieczkiMiejsca2
AS
    SELECT w.ID_WYCIECZKI, w.KRAJ, w.DATA, w.NAZWA, w.LICZBA_MIEJSC,
    w.LICZBA_WOLNYCH_MIEJSC
    FROM WYCIECZKI w
    LEFT JOIN (SELECT * FROM REZERWACJE r WHERE r.status <> 'A') r
    on r.ID_WYCIECZKI=w.ID_WYCIECZKI
    GROUP BY w.ID_WYCIECZKI, w.KRAJ, w.DATA, w.NAZWA,
    w.LICZBA_MIEJSC, w.LICZBA_WOLNYCH_MIEJSC;

```

## 7.4 MODYFIKACJA PROCEDUR POBIERAJĄCYCH

Poniżej umieszczono jedyną procedurę pobierającą dane, w której dokonano korekt ze względu na dodaną powyżej dodatkową kolumnę w tabeli WYCIECZKI. Zmiany jak powyżej dotyczyły uproszczenia składni dzięki użyciu nowego atrybutu

### 7.4.1 PROCEDURA DOSTĘPNE MIEJSCA 2

```

CREATE OR REPLACE FUNCTION DOSTEPNE_WYCIECZKI2
(

```

```

        kraj WYCIECZKI.KRAJ%TYPE, data_od DATE, data_do DATE
    )
RETURN WYCIECZKI_TABLE AS
    result WYCIECZKI_TABLE;
BEGIN
    IF data_od > data_do THEN
        RAISE_APPLICATION_ERROR(-20001, 'Data_od must be before
data_do');
    END IF;
    SELECT WYCIECZKA(w.ID_WYCIECZKI, w.NAZWA, w.KRAJ, w.DATA, w.OPIS,
w.LICZBA_MIEJSC)
    BULK COLLECT INTO result
    FROM WYCIECZKI w
    WHERE w.LICZBA_WOLNYCH_MIEJSC>0 and
w.KRAJ=DOSTEPNE_WYCIECZKI2.kraj and w.DATA >= data_od and w.DATA <=
data_do;
RETURN result;
END;

```

## 7.5 MODYFIKACJA PROCEDUR MODYFIKUJĄCYCH

### 7.5.1 PROCEDURA DODAJ REZERWACJE 2

Zmiana dokonana w tej procedurze została wyszczególniona w komentarzu w kodzie.

```

CREATE OR REPLACE PROCEDURE DODAJ_REZERWACJE2
(
    ID_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE,
    ID_osoby OSOBY.ID_OSOBY%TYPE
)
AS
    person_exists      INT;
    trip_available      INT;
    reservation_exists INT;
    new_reservation_ID INT;
BEGIN
    SELECT COUNT(*)
    INTO person_exists
    FROM OSOBY
    WHERE OSOBY.ID_OSOBY = DODAJ_REZERWACJE2.ID_osoby;
    IF person_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Person with chosen ID does not
exist.');
```

```

    END IF;

    SELECT COUNT(*)

```

```

    INTO trip_available
    FROM WycieczkiDostepne wd
    WHERE wd.ID_WYCIECZKI = DODAJ_REZERWACJE2.ID_wycieczki;

    IF trip_available = 0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID is not
available or does not exist.');
```

available or does not exist.');

```

    END IF;

    SELECT COUNT(*)
    INTO reservation_exists
    FROM REZERWACJE r
    WHERE r.ID_WYCIECZKI = DODAJ_REZERWACJE2.ID_wycieczki AND r.ID_OSOBY
= DODAJ_REZERWACJE2.ID_osoby;

    IF reservation_exists > 0 THEN
        RAISE_APPLICATION_ERROR(-20004, 'Reservation for chosen trip ID
and person ID has already existed.');
```

and person ID has already existed.');

```

    END IF;

    INSERT INTO REZERWACJE (ID_WYCIECZKI, ID_OSOBY, STATUS)
    VALUES (DODAJ_REZERWACJE2.ID_wycieczki, DODAJ_REZERWACJE2.ID_osoby,
'N')
    RETURNING NR_REZERWACJI INTO new_reservation_ID;

-- ADDED!
    UPDATE WYCIECZKI
    SET LICZBA_WOLNYCH_MIEJSC = LICZBA_WOLNYCH_MIEJSC - 1
    WHERE ID_WYCIECZKI = DODAJ_REZERWACJE2.ID_wycieczki;

    INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
    VALUES (new_reservation_ID, CURRENT_DATE, 'N');
COMMIT;
END;
```

SPRAWDZENIE POPRAWNOŚCI DZIAŁANIA:

```

BEGIN
    DODAJ_REZERWACJE2(4,1);
END;
```

Poprawnie dodano nową rezerwację do tabeli REZERWACJE

	NR_REZERWACJI	ID_WYCIECZKI	ID_OSOBY	STATUS
1	1	1	1	N
2	2	2	2	P
3	3	3	3	Z
4	4	4	4	A
5	5	1	5	A
6	6	2	6	Z
7	7	3	7	P
8	8	4	8	N
9	9	4	9	Z
10	10	2	11	A
11	21	4	11	N
12	41	4	1	N

Poprawnie dodano nową rezerwację do tabeli REZERWACJE\_LOG

	ID	ID_REZERWACJI	DATA	STATUS
1	1	21	2020-03-07 18:19:08	N
2	2	9	2020-03-07 18:36:33	Z
3	21	41	2020-03-09 19:22:17	N

Poprawnie zmniejszono liczbę wolnych miejsc o 1 na wycieczkę nr 4

	ID_WYCIECZKI	NAZWA	KRAJ	DATA	OPIS	LICZBA_MIEJSC	LICZBA_WOLNYCH_MIEJSC
1	1	Wycieczka do Paryża	Francja	2016-01-01 00:00:00	Ciekawa wycieczka ...	3	2
2	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Najciekawsza wycieczka ...	3	1
3	3	Wieliczka	Polska	2017-03-03 00:00:00	Zadziwiająca kopalnia ...	2	0
4	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Niesamowita podróż po ...	5	2

## 7.5.2 PROCEDURA ZMIEN STATUS REZERWACJI 2

Zmiana dokonana w tej procedurze została wyszczególniona w komentarzu w kodzie.

```
CREATE PROCEDURE ZMIEN_STATUS_REZERWACJI2
(
    ID_rezerwacji REZERWACJE.NR_REZERWACJI%TYPE,
    status REZERWACJE.STATUS%TYPE
)
AS
    not_updated_status REZERWACJE.STATUS%TYPE;
    reservation_exists INT;
BEGIN
    SELECT COUNT(*)
    INTO reservation_exists
    FROM RezerwacjeWPrzyszlosci rwp
    JOIN REZERWACJE r ON r.ID_WYCIECZKI=rwp.ID_WYCIECZKI
    JOIN OSOBY o ON r.ID_OSOBY = o.ID_OSOBY
    WHERE r.NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI2.ID_rezerwacji;
```



```

    IF reservation_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'There is no reservation with
given ID');
    END IF;

    SELECT STATUS
    INTO not_updated_status
    FROM REZERWACJE r
    WHERE r.NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI2.ID_REZERWACJI;
    CASE
        WHEN not_updated_status IS NULL
            THEN RAISE_APPLICATION_ERROR(-20002, 'Reservation with chosen
ID does not exist.');
```

```

        WHEN not_updated_status = 'A'
            THEN RAISE_APPLICATION_ERROR(-20003, 'Status of cancelled
reservation ("A") can not be changed.');
```

```

        WHEN not_updated_status = 'P'
            THEN
                IF ZMIEN_STATUS_REZERWACJI2.status NOT IN ('Z','A') THEN
                    RAISE_APPLICATION_ERROR(-20003,
                        'Status of confirmed reservation
("P") can be changed only to "Z" (confirmed and paid) or "A"
(cancelled).');
```

```

                END IF;

        WHEN not_updated_status = 'Z'
            THEN
                IF ZMIEN_STATUS_REZERWACJI2.status <> 'A' THEN
                    RAISE_APPLICATION_ERROR(-20003,
                        'Status of confirmed and paid
reservation ("Z") can be changed only' to "A" (cancelled).');
```

```

                END IF;

        ELSE
            RAISE_APPLICATION_ERROR(-30000, 'Internal application error,
CASE not found.');
```

```

        END CASE;

    UPDATE REZERWACJE
    SET STATUS = ZMIEN_STATUS_REZERWACJI2.status
    WHERE NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI2.ID_rezerwacji;

    -- ADDED!
```

```

IF status = 'A' THEN
    UPDATE WYCIECZKI w
    SET w.LICZBA_WOLNYCH_MIEJSC=w.LICZBA_WOLNYCH_MIEJSC+1
    WHERE w.ID_WYCIECZKI = (SELECT ID_WYCIECZKI
                             FROM REZERWACJE r
                             WHERE r.NR_REZERWACJI =
ZMIEN_STATUS_REZERWACJI2.ID_rezerwacji);
END IF;
INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
VALUES (ZMIEN_STATUS_REZERWACJI2.ID_rezerwacji, CURRENT_DATE,
ZMIEN_STATUS_REZERWACJI2.status);
COMMIT;
END;

```

SPRAWDZENIE POPRAWNOŚCI DZIAŁANIA:

```

BEGIN
    ZMIEN_STATUS_REZERWACJI2(9, 'A');
END;

```

Poprawnie zmieniono status rezerwacji w tabeli REZERWACJE

	NR_REZERWACJI	ID_WYCIECZKI	ID_OSOBY	STATUS
1	1	1	1	N
2	2	2	2	P
3	3	3	3	Z
4	4	4	4	A
5	5	1	5	A
6	6	2	6	Z
7	7	3	7	P
8	8	4	8	N
9	9	4	9	A
10	10	2	11	A
11	21	4	11	N
12	41	4	1	N

Poprawnie dodano wiersz do tabeli REZERWACJE\_LOG ze zmienionym statusem 'A'

	ID	ID_REZERWACJI	DATA	STATUS
1	1	21	2020-03-07 18:19:08	N
2	2	9	2020-03-07 18:36:33	Z
3	21	41	2020-03-09 19:22:17	N
4	23	9	2020-03-09 19:35:20	A

Poprawne zwiększenie liczby wolnych miejsc na wycieczkę numer 4

	ID_WYCIECZKI	NAZWA	KRAJ	DATA	OPIS	LICZBA_MIEJSC	LICZBA_WOLNYCH_MIEJSC
1	1	Wycieczka do Paryża	Francja	2016-01-01 00:00:00	Ciekawa wycieczka ...	3	2
2	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Najciekawsza wycieczka ...	3	1
3	3	Wieliczka	Polska	2017-03-03 00:00:00	Zadziwiająca kopalnia ...	2	0
4	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Niesamowita podróż po ...	6	3

### 7.5.3 PROCEDURA ZMIEN LICZBĘ MIEJSC 2

```

CREATE OR REPLACE PROCEDURE ZMIEN_LICZBE_MIEJSC2
(
    ID_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE,
    nowa_liczba_miejsc WYCIECZKI.LICZBA_MIEJSC%TYPE
)
AS
    reserved_places INT;
BEGIN
    SELECT wm.LICZBA_MIEJSC - wm.LICZBA_WOLNYCH_MIEJSC
    INTO reserved_places
    FROM WycieczkiMiejscs wm
    WHERE wm.ID_WYCIECZKI = ZMIEN_LICZBE_MIEJSC2.ID_wycieczki;

    IF reserved_places > nowa_liczba_miejsc OR nowa_liczba_miejsc < 0
    THEN
        RAISE_APPLICATION_ERROR(-20004, 'Given number of places is
        inappropriate (smaller than number of reserved places).');
    END IF;

    UPDATE WYCIECZKI
    SET LICZBA_MIEJSC = ZMIEN_LICZBE_MIEJSC2.nowa_liczba_miejsc,
    -- ADDED!
        LICZBA_WOLNYCH_MIEJSC = ZMIEN_LICZBE_MIEJSC2.nowa_liczba_miejsc -
        reserved_places
    WHERE ID_WYCIECZKI = ZMIEN_LICZBE_MIEJSC2.ID_wycieczki;

    COMMIT;
    EXCEPTION
        WHEN NO_DATA_FOUND THEN
            RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID not
            found. ');
    END;

```

SPRAWDZENIE POPRAWNOŚCI DZIAŁANIA:

```

BEGIN
    ZMIEN_LICZBE_MIEJSC2(4,8);
END;

```

Poprawnie zaktualizowano liczbę miejsc oraz obliczono nową liczbę wolnych miejsc

	ID_WYCIECZKI	NAZWA	KRAJ	DATA	OPIS	LICZBA_MIEJSC	LICZBA_WOLNYCH_MIEJSC
1	1	Wycieczka do Paryża	Francja	2016-01-01 00:00:00	Ciekawa wycieczka ...	3	2
2	2	Piękny Kraków	Polska	2017-02-03 00:00:00	Najciekawsza wycieczka ...	3	1
3	3	Wieliczka	Polska	2017-03-03 00:00:00	Zadziwiająca kopalnia ...	2	0
4	4	Cesarskie Chiny	Chiny	2020-06-02 00:00:00	Niesamowita podróż po ...	8	5

## 8. DODANIE TRIGGERÓW

Triggery uwzględniają aktualizowanie wartości w kolumnie LICZBA\_WOLNYCH\_MIEJSC oraz uwzględniają operacje w tabeli dziennikującej REZERWACJE\_LOG

### 8.1 TRIGGER DODAJ REZERWACJE

```
CREATE OR REPLACE TRIGGER DODANIE_REZERWACJI_TRIGGER
  AFTER INSERT
  ON REZERWACJE
  FOR EACH ROW
BEGIN
  INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
  VALUES (:NEW.NR_REZERWACJI, CURRENT_DATE, :NEW.STATUS);

  UPDATE WYCIECZKI w
  SET LICZBA_WOLNYCH_MIEJSC = LICZBA_WOLNYCH_MIEJSC - 1
  WHERE w.ID_WYCIECZKI = :NEW.ID_WYCIECZKI;
END;
```

### 8.2 TRIGGER ZMIANA STATUSU REZERWACJI

```
CREATE OR REPLACE TRIGGER ZMIANA_STATUSU_TRIGGER
  AFTER UPDATE OF STATUS
  ON REZERWACJE
  FOR EACH ROW
BEGIN
  INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
  VALUES (:NEW.NR_REZERWACJI, CURRENT_DATE, :NEW.STATUS);

  IF :NEW.STATUS = 'A' THEN
    UPDATE WYCIECZKI w
    SET LICZBA_WOLNYCH_MIEJSC = LICZBA_WOLNYCH_MIEJSC + 1
    WHERE w.ID_WYCIECZKI = :NEW.ID_WYCIECZKI;
  END IF;
END;
```

## 8.3 TRIGGER ZMIANA LICZBY MIEJSC

```
CREATE OR REPLACE TRIGGER ZMIANA_LICZBY_MIEJSC_TRIGGER
  AFTER UPDATE OF LICZBA_MIEJSC
  ON WYCIECZKI
  FOR EACH ROW
BEGIN
  UPDATE WYCIECZKI

  SET LICZBA_WOLNYCH_MIEJSC = LICZBA_WOLNYCH_MIEJSC +
  (:NEW.LICZBA_MIEJSC - LICZBA_MIEJSC);
END;
```

## 8.4 TRIGGER ZAKAZ USUWANIA REZERWACJI

```
CREATE OR REPLACE TRIGGER ZAKAZ_USUWANIA_REZERWACJI
  BEFORE DELETE
  ON REZERWACJE
  FOR EACH ROW
BEGIN
  raise_application_error(-20008, 'Removing reservation is
  forbidden.');
```

SPRAWDZENIE POPRAWNOŚCI DZIAŁANIA:

```
DELETE FROM REZERWACJE
WHERE NR_REZERWACJI=41;
```

```
[72000][20008] ORA-20008: Removing reservation forbidden.
ORA-06512: przy "BD_NOWARA.ZABRON_USUWANIA_REZERWACJI", linia 2
ORA-04088: błąd w trakcie wykonywania wyzwalacza "BD_NOWARA.ZABRON_USUWANIA_REZERWACJI"
Position: 12
```

## 8.5 DRUGA MODYFIKACJA PROCEDURY DODAJ REZERWACJE

Zmiana dokonana w tej procedurze została wyszczególniona w komentarzu w kodzie.

```
CREATE OR REPLACE PROCEDURE DODAJ_REZERWACJE3
(
  ID_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE,
  ID_osoby OSOBY.ID_OSOBY%TYPE
```

```

    )
AS
    person_exists      INT;
    trip_available      INT;
    reservation_exists INT;
BEGIN
    SELECT COUNT(*)
    INTO person_exists
    FROM OSOBY
    WHERE OSOBY.ID_OSOBY = DODAJ_REZERWACJE3.ID_osoby;
    IF person_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Person with chosen ID does not
exist.');
```

```

    END IF;

    SELECT COUNT(*)
    INTO trip_available
    FROM WycieczkiDostepne wd
    WHERE wd.ID_WYCIECZKI = DODAJ_REZERWACJE3.ID_wycieczki;

    IF trip_available = 0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID is not
available or does not exist.');
```

```

    END IF;

    SELECT COUNT(*)
    INTO reservation_exists
    FROM REZERWACJE r
    WHERE r.ID_WYCIECZKI = DODAJ_REZERWACJE3.ID_wycieczki AND r.ID_OSOBY
= DODAJ_REZERWACJE3.ID_osoby;

    IF reservation_exists > 0 THEN
        RAISE_APPLICATION_ERROR(-20004, 'Reservation for chosen trip ID
and person ID has already existed.');
```

```

    END IF;
-- CHANGED! We don't need now new_reservation_ID for table
RESERVATION_LOG, because our trigger DODAJ_REZERWACJE_TRIGGER solve this
problem
    INSERT INTO REZERWACJE (ID_WYCIECZKI, ID_OSOBY, STATUS)
    VALUES (DODAJ_REZERWACJE3.ID_wycieczki, DODAJ_REZERWACJE3.ID_osoby,
'N');
```

```

COMMIT;
END;
```

## 8.6 DRUGA MODYFIKACJA PROCEDURY ZMIEN STATUS REZERWACJI

Zmiana dokonana w tej procedurze została wyszczególniona w komentarzu w kodzie.

```
CREATE PROCEDURE ZMIEN_STATUS_REZERWACJI3
(
    ID_rezerwacji REZERWACJE.NR_REZERWACJI%TYPE,
    status REZERWACJE.STATUS%TYPE
)
AS
    not_updated_status REZERWACJE.STATUS%TYPE;
    reservation_exists INT;
BEGIN
    SELECT COUNT(*)
    INTO reservation_exists
    FROM RezerwacjeWPrzyszlosci rwp
    JOIN REZERWACJE r ON r.ID_WYCIECZKI=rwp.ID_WYCIECZKI
    JOIN OSOBY o ON r.ID_OSOBY = o.ID_OSOBY
    WHERE r.NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI3.ID_rezerwacji;

    IF reservation_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'There is no reservation with
given ID');
    END IF;

    SELECT STATUS
    INTO not_updated_status
    FROM REZERWACJE r
    WHERE r.NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI3.ID_REZERWACJI;
    CASE
        WHEN not_updated_status IS NULL
            THEN RAISE_APPLICATION_ERROR(-20002, 'Reservation with chosen
ID does not exist.');
```

```
        WHEN not_updated_status = 'A'
            THEN RAISE_APPLICATION_ERROR(-20003, 'Status of cancelled
reservation ("A") can not be changed.');
```

```
        WHEN not_updated_status = 'P'
            THEN
```

```

        IF ZMIEN_STATUS_REZERWACJI3.status NOT IN ('Z','A') THEN
            RAISE_APPLICATION_ERROR(-20003,
                'Status of confirmed reservation
("P") can be changed only to "Z" (confirmed and paid) or "A"
(cancelled).');
        END IF;

        WHEN not_updated_status = 'Z'
        THEN
            IF ZMIEN_STATUS_REZERWACJI3.status <> 'A' THEN
                RAISE_APPLICATION_ERROR(-20003,
                    'Status of confirmed and paid
reservation ("Z") can be changed only to "A" (cancelled).');
            END IF;

        ELSE
            RAISE_APPLICATION_ERROR(-30000, 'Internal application error,
CASE not found.');
```

*-- CHANGED!; We don't need now to update in procedure  
LICZBA\_WOLNYCH\_MIEJSC because our trigger ZMIANA\_STATUSU\_TRIGGEER solve  
this problem*

```

        COMMIT;
    END;

    UPDATE REZERWACJE
    SET STATUS = ZMIEN_STATUS_REZERWACJI3.status
    WHERE NR_REZERWACJI = ZMIEN_STATUS_REZERWACJI3.ID_rezerwacji;

```

## 8.7 DRUGA MODYFIKACJA PROCEDURY ZMIEN LICZBĘ MIEJSC

Zmiana dokonana w tej procedurze została wyszczególniona w komentarzu w kodzie.

```

CREATE OR REPLACE PROCEDURE ZMIEN_LICZBE_MIEJSC3
(
    ID_wycieczki WYCIECZKI.ID_WYCIECZKI%TYPE,
    nowa_liczba_miejsc WYCIECZKI.LICZBA_MIEJSC%TYPE
)
AS
    reserved_places INT;
BEGIN

```



```

SELECT wm.LICZBA_MIEJSC - wm.LICZBA_WOLNYCH_MIEJSC
INTO reserved_places
FROM WycieczkiMiejsca wm
WHERE wm.ID_WYCIECZKI = ZMIEN_LICZBE_MIEJSC3.ID_wycieczki;

IF reserved_places > nowa_liczba_miejsc OR nowa_liczba_miejsc<0
THEN
    RAISE_APPLICATION_ERROR(-20004,'Given number of places is
inappropriate (smaller than number of reserved places).');
END IF;

UPDATE WYCIECZKI
SET LICZBA_MIEJSC = ZMIEN_LICZBE_MIEJSC3.nowa_liczba_miejsc
-- CHANGED! We don't need now to update in procedure
LICZBA_WOLNYCH_MIEJSC because our trigger ZMIANA_LICZBY_MIEJSC_TRIGGEER
solve this problem
WHERE ID_WYCIECZKI = ZMIEN_LICZBE_MIEJSC3.ID_wycieczki;
COMMIT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID not
found. ');
END;

```