# Bazy danych Oracle PL/SQL,- ćwiczenie – Karolina Biela

## Tworzenie tabel według skryptu podanego w opisie ćwiczenia

## Uzupełnianie tabel

### Uzupełnianie tabeli OSOBY za pomocą generatora danych

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

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Roz', 'Micka', '88344292161', 'rmicka0@state.tx.us');

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Tomasina', 'Heisham', '91118886174', 'theisham1@ebay.com');

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Franny', 'McIlwraith', '61116594248', 'fmcilwraith2@mashable.com');

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Ginelle', 'Shalders', '17541468241', 'gshalders3@vk.com');

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Brodie', 'Ech', '80250944218', 'bech4@whitehouse.gov');

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Errol', 'Woolf', '64727894728', 'ewoolf5@hc360.com');

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Drusie', 'Steynor', '24698681028', 'dsteynor6@mayoclinic.com');

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Aristotle', 'Leggate', '70734501079', 'aleggate7@apache.org');

insert into OSOBY (imie, nazwisko, pesel, kontakt) values ('Crissie', 'Pirozzi', '16329833919', 'cpirozzi8@over-blog.com');



### Uzupełnianie tabeli WYCIECZKI

INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba\_miejsc)

VALUES ('Wycieczka do Paryza','Francja','2016-01-01','Ciekawa wycieczka ...',3);

INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba\_miejsc)

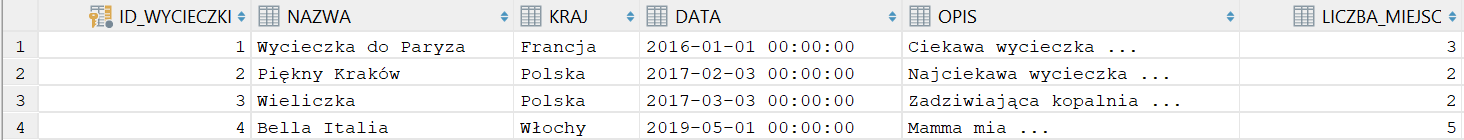
VALUES ('Piękny Kraków','Polska','2017-02-03','Najciekawa wycieczka ...',2);

INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba\_miejsc)

VALUES ('Wieliczka','Polska','2017-03-03','Zadziwiająca kopalnia ...',2);

INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba\_miejsc)

VALUES ('Bella Italia','Włochy','2019-05-01','Mamma Mia ...',5);



### Uzupełnianie tabeli REZERWACJE

INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (1,1,'N');

INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (2,2,'P');

INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (1,5,'Z');

INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (4,3,'A');

INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (3,8,'N');

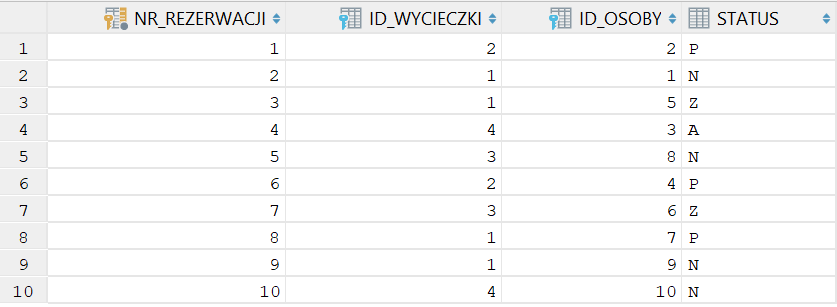
INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (2,4,'P');

INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (3,6,'Z');

INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (1,7,'P');

INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (1,9,'N');

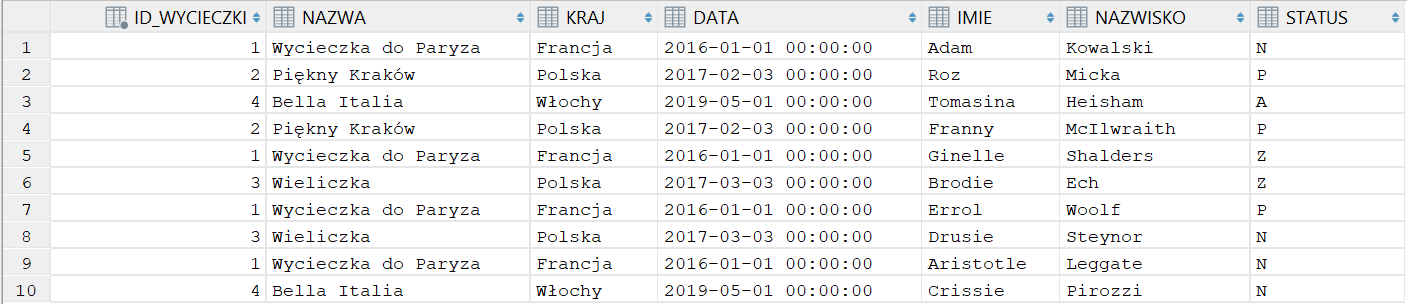
INSERT INTO rezerwacje(id\_wycieczki, id\_osoby, status) VALUES (4,10,'N');



## Tworzenie widoków

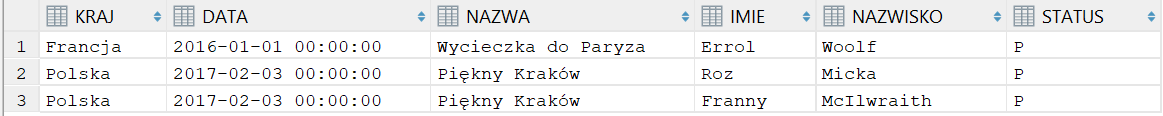
### wycieczki\_osoby

**CREATE VIEW** wycieczki\_osoby  
 **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**;



### wycieczki\_osoby\_potwierdzone

**create or replace view** WYCIECZKI\_OSOBY\_POTWIERDZONE **as  
 SELECT  
 KRAJ**, **DATA**, **NAZWA**, **IMIE**, **NAZWISKO**, **STATUS  
 FROM** WYCIECZKI\_OSOBY  
**where** r.status = **'P'**



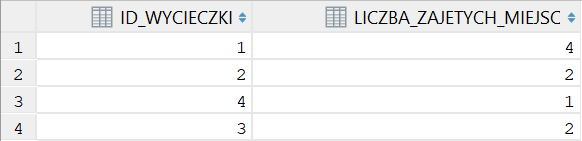
### wycieczki\_przyszle

**create or replace view** WYCIECZKI\_PRZYSZLE **as  
SELECT  
 KRAJ**,**DATA**,**NAZWA**,**IMIE**,**NAZWISKO**,**STATUS  
 FROM** WYCIECZKI\_OSOBY  
**where STATUS** = **'N' and DATA** > *SYSDATE*;

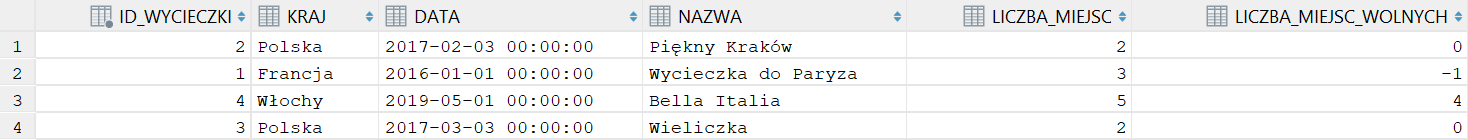


**WIDOK POMOCNICZY**

**create view** miejsca\_zajęte  
**as  
select** R.**ID\_WYCIECZKI**,  
 *count*(R.**ID\_WYCIECZKI**) **AS** LICZBA\_ZAJETYCH\_MIEJSC  
**FROM** REZERWACJE R  
**WHERE** R.**STATUS** !=**'A'  
Group by** R.**ID\_WYCIECZKI**;



### wycieczki\_miejsca

**create or replace view** WYCIECZKI\_MIEJSCA **as  
 SELECT distinct** w.**ID\_WYCIECZKI**,  
 w.**KRAJ**,  
 w.**DATA**,  
 w.**NAZWA**,  
 w.**LICZBA\_MIEJSC**,  
 (w.**LICZBA\_MIEJSC** - (**select** *count*(R.**ID\_WYCIECZKI**) **AS** LICZBA\_ZAJETYCH\_MIEJSC **FROM** REZERWACJE R  
 **WHERE** R.**STATUS** !=**'A' and** w.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI  
 Group by** R.**ID\_WYCIECZKI**)) **as** liczba\_miejsc\_wolnych  
  
 **FROM** WYCIECZKI w  
 **JOIN** REZERWACJE r **ON** w.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI  
 JOIN** OSOBY o **ON** r.**ID\_OSOBY** = o.**ID\_OSOBY**

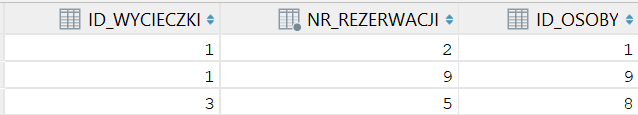
### dostępne\_wycieczki

**create or replace view** DOSTEPNE\_WYCIECZKI **as  
select KRAJ**, **DATA**, **NAZWA**, **LICZBA\_MIEJSC**, **ID\_WYCIECZKI  
FROM** WYCIECZKI\_MIEJSCA  
 **WHERE LICZBA\_MIEJSC\_WOLNYCH**>0 **AND DATA** >*SYSDATE*;



### rezerwacje\_do\_anulowania

**create view** REZREWACJE\_DO\_ANULOWANIA **as  
 SELECT**r.**ID\_WYCIECZKI**,  
r.**NR\_REZERWACJI**,  
r.**ID\_OSOBY  
 FROM** WYCIECZKI w  
 **JOIN** REZERWACJE r **ON** w.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI  
 JOIN** OSOBY o **ON** r.**ID\_OSOBY** = o.**ID\_OSOBY  
 where** r.**STATUS**=**'N' and** *SYSDATE* + 7 >=w.**DATA**



## PROCEDURY, FUNKCJE

### Procedury pomocnicze

**create or replace PROCEDURE** *CZY\_WYCIECZKA\_ISTNIEJE* (IDWYCIECZKI **IN NUMBER**) **AS** L\_WYCIECZEK **NUMBER**;  
 D\_WYCIECZKI **DATE**;  
 **BEGIN  
 SELECT** *COUNT*(**ID\_WYCIECZKI**) **INTO** L\_WYCIECZEK **FROM** WYCIECZKI **WHERE ID\_WYCIECZKI**=IDWYCIECZKI;  
 **IF** L\_WYCIECZEK = 0 **THEN** *raise\_application\_error* (-20001,**'WYCIECZKA O PODANYM ID NIE ISTNIEJE'**);  
 **END IF** ;  
  
 **SELECT DATA INTO** D\_WYCIECZKI **FROM** WYCIECZKI **WHERE** IDWYCIECZKI=**ID\_WYCIECZKI**;  
 **IF** d\_WYCIECZKI < *SYSDATE* **THEN** *raise\_application\_error* (-20001,**'WYCIECZKA O PODANYM ID JUŻ SIĘ ODBYŁA'**);  
 **END IF** ;  
 **END**;

**declare** IDWYCIECZKI **NUMBER** := 0;  
**begin** *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI => IDWYCIECZKI);  
**end**;



**create PROCEDURE** CZY\_OSOBA\_ISTNIEJE(IDOSOBY **NUMBER**) **AS** L\_OSOBY **NUMBER**;  
 **BEGIN  
 SELECT** *COUNT*(ID\_OSOBY) **INTO** L\_OSOBY **FROM** OSOBY **WHERE** ID\_OSOBY=IDOSOBY;  
 IF L\_OSOBY = 0 **THEN** raise\_application\_error (-20001,**'OSOBA O PODANYM ID NIE ISTNIEJE'**);  
 **END** IF;  
 **END**;

**declare** IDOSOBY **NUMBER** := 0;  
**begin** *CZY\_OSOBA\_ISTNIEJE*(IDOSOBY => IDOSOBY);  
**end**; 

**create OR** REPLACE **PROCEDURE** LICZBA\_WOLNYCH\_MIEJSC (IDWYCIECZKI **IN NUMBER** ) **AS**L\_WYCIECZEK **NUMBER**;  
**BEGIN  
 SELECT** *COUNT*(**ID\_WYCIECZKI**) **INTO** L\_WYCIECZEK **FROM** DOSTEPNE\_WYCIECZKI **WHERE ID\_WYCIECZKI**=IDWYCIECZKI;  
  
 IF L\_WYCIECZEK = 0 **THEN** raise\_application\_error (-20001,**'PODANA WYCIECZKA JEST NIEDOSTĘPNA'**);  
 **END** IF;  
**END**;

**declare** IDWYCIECZKI **NUMBER** := 0;  
**begin** *LICZBA\_WOLNYCH\_MIEJSC*(IDWYCIECZKI => IDWYCIECZKI);  
**end**;



**create TYPE** pojedyncza\_rezerwacja **AS OBJECT**(  
ID\_OSOBY **NUMBER**,  
IMIE **VARCHAR2**(50),  
NAZWISKO **VARCHAR2**(50),  
STATUS **CHAR**(1),  
ID\_WYCIECZKI **NUMBER**,  
NAZWA **VARCHAR2**(100),  
KRAJ **VARCHAR2**(50),  
**DATA DATE**);

**CREATE OR REPLACE TYPE** REZERWACJE\_TABELA **AS TABLE OF pojedyncza\_rezerwacja**;

### uczestnicy\_wycieczki

**create OR REPLACE FUNCTION** *UCZESTNICY\_WYCIECZKI* (IDWYCIECZKI **NUMBER**)  
**RETURN REZERWACJE\_TABELA IS** licznik **PLS\_INTEGER** :=1;  
 pr **REZERWACJE\_TABELA** := **REZERWACJE\_TABELA**();  
 **CURSOR** CUR\_UCZESTNICY\_WYCIECZKI **IS  
 SELECT** w.**ID\_WYCIECZKI**,  
 w.**NAZWA**,  
 w.**KRAJ**,  
 w.**DATA**,  
 w.**IMIE**,  
 w.**NAZWISKO**,  
 w.**STATUS  
 FROM** wycieczki\_osoby w  
 **WHERE** w.**ID\_WYCIECZKI** = IDWYCIECZKI **AND** W.**STATUS**!=**'A'**;  
  
 **BEGIN  
 BEGIN** *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI);  
 **END**;  
 **FOR** i **IN** CUR\_UCZESTNICY\_WYCIECZKI **LOOP** pr.*EXTEND*;  
 pr(licznik) :=  
 **POJEDYNCZA\_REZERWACJA**(i.**IMIE**,i.**NAZWISKO**,i.**STATUS**,i.**ID\_WYCIECZKI**,i.**NAZWA**,i.**KRAJ**,i.**DATA**);  
 licznik := licznik+1;  
 **END LOOP**;  
 **RETURN** pr;  
 **END**;

**declare  
 result REZERWACJE\_TABELA**;  
 IDWYCIECZKI **NUMBER** := 4;  
**begin  
 result** := *UCZESTNICY\_WYCIECZKI*(IDWYCIECZKI => IDWYCIECZKI);  
 **open** ? **for select** *\** **from table**(**result**);  
**end**;



### rezerwacje\_osoby

**create OR REPLACE FUNCTION** *REZERWACJE\_OSOBY* (IDOSOBY **NUMBER**)  
**RETURN REZERWACJE\_TABELA IS** licznik **PLS\_INTEGER** :=1;  
 pr **REZERWACJE\_TABELA** := **REZERWACJE\_TABELA**();  
 **CURSOR** CUR\_REZERWACJE\_OSOBY **IS  
 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  
 WHERE** O.**ID\_OSOBY**=IDOSOBY;  
  
 **BEGIN  
 BEGIN** *CZY\_OSOBA\_ISTNIEJE*(IDOSOBY);  
 **END**;  
 **FOR** i **IN** CUR\_REZERWACJE\_OSOBY **LOOP** pr.*EXTEND*;  
 pr(licznik) :=  
 **POJEDYNCZA\_REZERWACJA**(i.**IMIE**,i.**NAZWISKO**,i.**STATUS**,i.**ID\_WYCIECZKI**,i.**NAZWA**,i.**KRAJ**,i.**DATA**);  
 licznik := licznik+1;  
 **END LOOP**;  
 **RETURN** pr;  
 **END**;

**declare  
 result REZERWACJE\_TABELA**;  
 IDOSOBY **NUMBER** := 1;  
**begin  
 result** := *REZERWACJE\_OSOBY*(IDOSOBY => IDOSOBY);  
 **open** ? **for select** *\** **from table**(**result**);  
**end**;



### przyszle\_rezerwacje\_osoby

**create OR REPLACE FUNCTION** *PRZYSZLE\_REZERWACJE\_OSOBY* (IDOSOBY **NUMBER**)  
**RETURN REZERWACJE\_TABELA IS** licznik **PLS\_INTEGER** :=1;  
 pr **REZERWACJE\_TABELA** := **REZERWACJE\_TABELA**();  
 **CURSOR** CUR\_PRZYSZLE\_REZERWACJE\_OSOBY **IS  
 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  
 WHERE** O.**ID\_OSOBY**=IDOSOBY **AND** *SYSDATE*<W.**DATA AND** R.**STATUS**!=**'A'**;  
  
 **BEGIN  
 BEGIN** *CZY\_OSOBA\_ISTNIEJE*(IDOSOBY);  
 **END**;  
 **FOR** i **IN** CUR\_PRZYSZLE\_REZERWACJE\_OSOBY **LOOP** pr.*EXTEND*;  
 pr(licznik) :=  
 **POJEDYNCZA\_REZERWACJA**(i.**IMIE**,i.**NAZWISKO**,i.**STATUS**,i.**ID\_WYCIECZKI**,i.**NAZWA**,i.**KRAJ**,i.**DATA**);  
 licznik := licznik+1;  
 **END LOOP**;  
 **RETURN** pr;  
 **END**;

**declare  
 result REZERWACJE\_TABELA**;  
 IDOSOBY **NUMBER** := 10;  
**begin  
 result** := *PRZYSZLE\_REZERWACJE\_OSOBY*(IDOSOBY => IDOSOBY);  
 **open** ? **for select** *\** **from table**(**result**);  
**end**;



### dostępne\_wycieczki

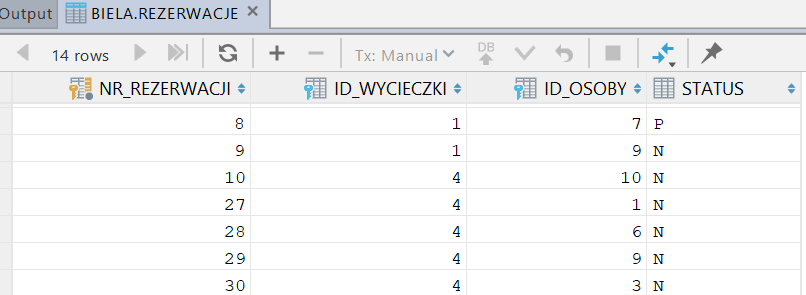
**create OR REPLACE FUNCTION** *DOSTEPNE\_WYCIECZKI\_FUNKCJA* (Z\_KRAJ **VARCHAR2**,OD\_DATA **DATE**, DO\_DATA **DATE**)  
**RETURN WYCIECZKI\_TABELA IS** licznik **PLS\_INTEGER** :=1;  
 pw **WYCIECZKI\_TABELA**:= **WYCIECZKI\_TABELA**();  
 **CURSOR** CUR\_DOSTEPNE\_WYCIECZKI\_FUNKCJA **IS  
  
 select KRAJ**, **DATA**, **NAZWA**, **LICZBA\_MIEJSC**, **ID\_WYCIECZKI  
 FROM** WYCIECZKI\_MIEJSCA  
 **WHERE LICZBA\_MIEJSC\_WOLNYCH**>0 **AND DATA**<=DO\_DATA **AND DATA**>=OD\_DATA **AND KRAJ**=Z\_KRAJ;  
  
 **BEGIN  
 FOR** i **IN** CUR\_DOSTEPNE\_WYCIECZKI\_FUNKCJA **LOOP** pw.*EXTEND*;  
 pw(licznik) :=  
 **POJEDYNCZA\_WYCIECZKA**(i.**KRAJ**,i.**DATA**,i.**NAZWA**,i.**LICZBA\_MIEJSC**,i.**ID\_WYCIECZKI**);  
 licznik := licznik+1;  
 **END LOOP**;  
 **RETURN** pw;  
 **END**;

**declare  
 result WYCIECZKI\_TABELA**;  
 Z\_KRAJ **VARCHAR2**(4000) := **'Włochy'**;  
 OD\_DATA **DATE** := *sysdate*;  
 DO\_DATA **DATE** := *sysdate*+10000;  
**begin  
 result** := *DOSTEPNE\_WYCIECZKI\_FUNKCJA*(  
 Z\_KRAJ => Z\_KRAJ,  
 OD\_DATA => OD\_DATA,  
 DO\_DATA => DO\_DATA  
 );  
 **open** ? **for select** *\** **from table**(**result**);  
**end**;



### dodaj\_rezerwacje

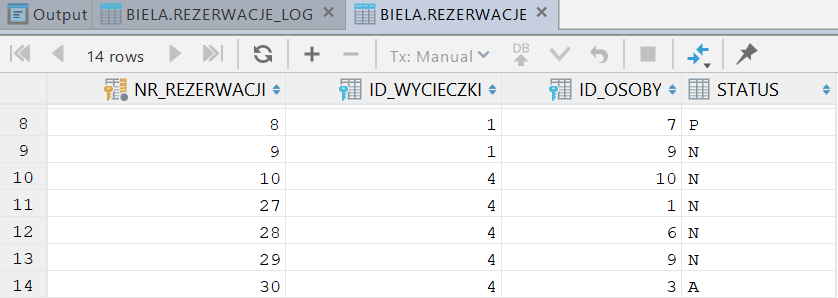
**CREATE OR REPLACE PROCEDURE** *DODAJ\_REZERWACJE* (IDWYCIECZKI **NUMBER**, IDOSOBY **NUMBER**) **AS** data\_wycieczki **DATE**;  
**BEGIN  
 BEGIN** *CZY\_OSOBA\_ISTNIEJE*(IDOSOBY);  
 *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI);  
 *LICZBA\_WOLNYCH\_MIEJSC*(IDWYCIECZKI);  
 **END**;  
  
 **SELECT DATA INTO** data\_wycieczki **FROM** WYCIECZKI  
 **WHERE** IDWYCIECZKI=**ID\_WYCIECZKI**;  
 **IF** data\_wycieczki < *SYSDATE* **THEN** *raise\_application\_error* (-20001,**'WYCIECZKA NIE JEST DOSTĘPNA'**);  
 **end if**;  
 **INSERT INTO** REZERWACJE(**ID\_WYCIECZKI**,**ID\_OSOBY**,**STATUS**) **VALUES** (IDWYCIECZKI,IDOSOBY,**'N'**);  
**END**;



1. zmien\_status\_rezerwacji

**CREATE OR REPLACE PROCEDURE** *ZMIEN\_STATUS\_REZERWACJI* (NRREZERWACJI **NUMBER**, NOWY\_STATUS **CHAR**) **AS** stary\_status **CHAR**;  
**BEGIN  
 BEGIN** *CZY\_REZERWACJA\_ISTNIEJE*(NRREZERWACJI);  
 **END**;  
 **SELECT STATUS INTO** stary\_status **FROM** REZERWACJE **WHERE** NRREZERWACJI=**NR\_REZERWACJI**;  
 **IF** stary\_status =**'A' THEN** *raise\_application\_error* (-20001,**'REZERWACJA ZOSTALA ANULOWANA'**);  
 **ELSIF** stary\_status=NOWY\_STATUS **THEN** *raise\_application\_error* (-20001,**'STATUS REZERWACJI JEST TAKI SAM'**);  
 **ELSIF** stary\_status=**'Z' AND** (NOWY\_STATUS=**'N' OR** NOWY\_STATUS=**'P'**) **THEN** *raise\_application\_error* (-20001,**'REZERWACJA JEST ZAPŁACONA I POTWIERDZONA, NIE MOŻNA WPROWADZIC TAKIEJ ZMIANY'**);  
 **END IF**;  
  
 **UPDATE** REZERWACJE **SET STATUS** = NOWY\_STATUS **WHERE NR\_REZERWACJI** = NRREZERWACJI;  
**END**;

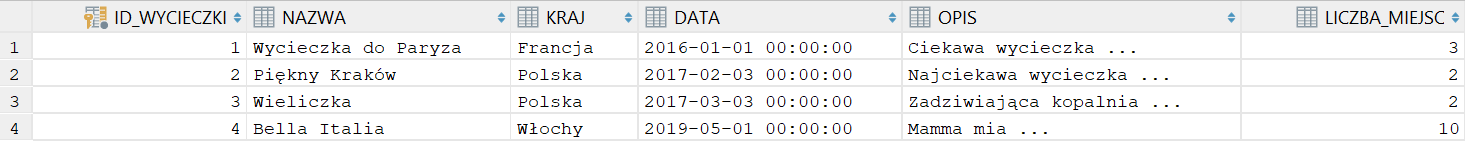
Zmiana statusu rezerwacji o numerze 30 na anulowaną.



1. zmien\_liczbe\_miejsc

**CREATE OR REPLACE PROCEDURE** *ZMIEN\_LICZBE\_MIEJSC* (IDWYCIECZKI **NUMBER**, NOWA\_LICZBA\_MIEJSC **NUMBER**) **AS** liczba\_zarezerwowanych\_miejsc **number**;  
**BEGIN  
 BEGIN** *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI);  
 **END**;  
 **SELECT liczba\_zajetych\_miejsc INTO** liczba\_zarezerwowanych\_miejsc **FROM** MIEJSCA\_ZAJĘTE **WHERE ID\_WYCIECZKI**=IDWYCIECZKI;  
 **IF** liczba\_zarezerwowanych\_miejsc > NOWA\_LICZBA\_MIEJSC **THEN** *raise\_application\_error* (-20001,**'NIE MOŻNA ZMIEJSZYC LICZBY MIEJSC Z POWODU ILOSCI REZERWACJI'**);  
 **END IF**;  
  
 **UPDATE** WYCIECZKI **SET LICZBA\_MIEJSC** = NOWA\_LICZBA\_MIEJSC **WHERE** IDWYCIECZKI = **ID\_WYCIECZKI**;  
  
**END**;

Zmiana liczby miejsc z 5 na 10.



# Zadanie 6

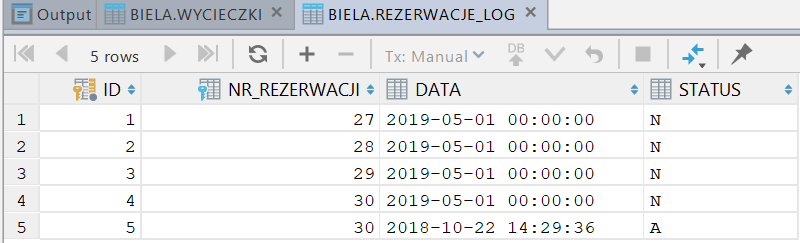
## rezerwacje\_log

**create table** REZERWACJE\_LOG  
(  
 **ID NUMBER GENERATED ALWAYS AS IDENTITY NOT NULL**,  
 **NR\_REZERWACJI NUMBER**,  
 **DATA DATE**,  
 **STATUS CHAR**(1),  
 **CONSTRAINT** REZERWACJE\_LOG\_PK **PRIMARY KEY** (**ID**)  
);

## modyfikacje procedur

**CREATE OR REPLACE PROCEDURE** *ZMIEN\_STATUS\_REZERWACJI* (NRREZERWACJI **NUMBER**, NOWY\_STATUS **CHAR**) **AS** stary\_status **CHAR**;  
  
**BEGIN  
 BEGIN** *CZY\_REZERWACJA\_ISTNIEJE*(NRREZERWACJI);  
 **END**;  
 **SELECT STATUS INTO** stary\_status **FROM** REZERWACJE **WHERE** NRREZERWACJI=**NR\_REZERWACJI**;  
 **IF** stary\_status =**'A' THEN** *raise\_application\_error* (-20001,**'REZERWACJA ZOSTALA ANULOWANA'**);  
 **ELSIF** stary\_status=NOWY\_STATUS **THEN** *raise\_application\_error* (-20001,**'STATUS REZERWACJI JEST TAKI SAM'**);  
 **ELSIF** stary\_status=**'Z' AND** (NOWY\_STATUS=**'N' OR** NOWY\_STATUS=**'P'**) **THEN** *raise\_application\_error* (-20001,**'REZERWACJA JEST ZAPŁACONA I POTWIERDZONA, NIE MOŻNA WPROWADZIC TAKIEJ ZMIANY'**);  
 **END IF**;  
  
 **UPDATE** REZERWACJE **SET STATUS** = NOWY\_STATUS **WHERE NR\_REZERWACJI** = NRREZERWACJI;  
 **INSERT INTO** REZERWACJE\_LOG(**NR\_REZERWACJI**,**DATA**,**STATUS**) **VALUES** (NRREZERWACJI,*SYSDATE*,NOWY\_STATUS);  
  
**END**;

Zapis zmiany status rezerwacji numer 30.



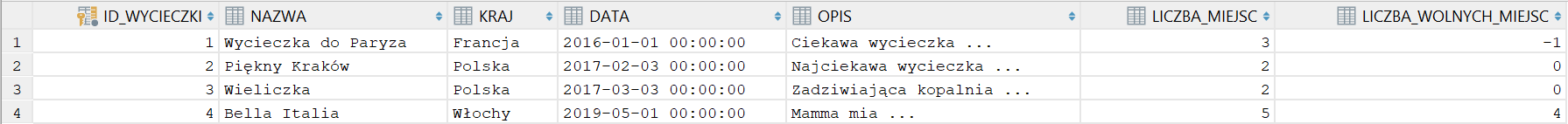
**CREATE OR REPLACE PROCEDURE** *DODAJ\_REZERWACJE* (IDWYCIECZKI **NUMBER**, IDOSOBY **NUMBER**) **AS** data\_wycieczki **DATE**;  
 id\_nowej\_rezerwacji **NUMBER**;  
**BEGIN  
 BEGIN** *CZY\_OSOBA\_ISTNIEJE*(IDOSOBY);  
 *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI);  
 *LICZBA\_WOLNYCH\_MIEJSC*(IDWYCIECZKI);  
 **END**;  
  
 **SELECT DATA INTO** data\_wycieczki **FROM** WYCIECZKI  
 **WHERE** IDWYCIECZKI=**ID\_WYCIECZKI**;  
 **IF** data\_wycieczki < *SYSDATE* **THEN** *raise\_application\_error* (-20001,**'WYCIECZKA NIE JEST DOSTĘPNA'**);  
 **end if**;  
 **INSERT INTO** REZERWACJE(**ID\_WYCIECZKI**,**ID\_OSOBY**,**STATUS**) **VALUES** (IDWYCIECZKI,IDOSOBY,**'N'**)  
 **RETURNING NR\_REZERWACJI INTO** id\_nowej\_rezerwacji;  
  
 **INSERT INTO** REZERWACJE\_LOG(**NR\_REZERWACJI**,**DATA**,**STATUS**) **VALUES** (id\_nowej\_rezerwacji,data\_wycieczki,**'N'**);  
**END**;  
/

## Dodawanie nowego pola do tabeli WYCIECZKI

**ALTER TABLE** WYCIECZKI **ADD LICZBA\_WOLNYCH\_MIEJSC NUMBER NULL**;

## Procedura przelicz

**CREATE OR REPLACE PROCEDURE** *przelicz* **AS  
BEGIN  
 UPDATE** WYCIECZKI W  
 **SET LICZBA\_WOLNYCH\_MIEJSC** = W.**LICZBA\_MIEJSC**-(**select** *count*(*\**)  
 **FROM** REZERWACJE R  
 **WHERE** R.**STATUS** !=**'A' and** W.**ID\_WYCIECZKI** = R.**ID\_WYCIECZKI**);  
**END**;



## Modyfikacje zestawu widoków

**create view** WYCIECZKI\_MIEJSCA2 **as  
 SELECT distinct** w.**ID\_WYCIECZKI**,  
 w.**KRAJ**,  
 w.**DATA**,  
 w.**NAZWA**,  
 w.**LICZBA\_MIEJSC**,  
 w.**LICZBA\_WOLNYCH\_MIEJSC  
 FROM** WYCIECZKI w

**create view** DOSTEPNE\_WYCIECZKI\_2 **as  
 select KRAJ**, **DATA**, **NAZWA**, **LICZBA\_MIEJSC**, **ID\_WYCIECZKI  
FROM** WYCIECZKI\_MIEJSCA2  
 **WHERE LICZBA\_WOLNYCH\_MIEJSC**>0 **AND DATA** >*SYSDATE*

## Modyfikacja procedur

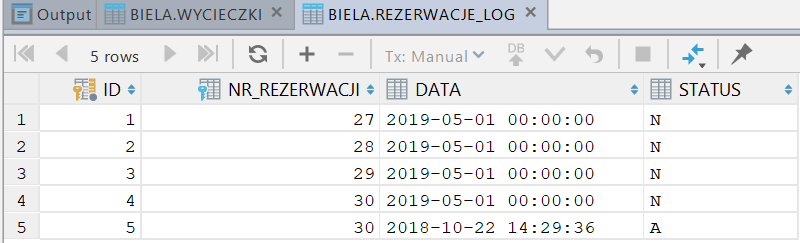
Modyfikacja procedury LICZBA\_WOLNYCH\_MIEJSC, poprzez użycie zmodyfikowanego widoku.

**create OR REPLACE PROCEDURE** *LICZBA\_WOLNYCH\_MIEJSC\_2* (IDWYCIECZKI **NUMBER** ) **AS**L\_WYCIECZEK **NUMBER**;  
**BEGIN  
SELECT** *COUNT*(**ID\_WYCIECZKI**) **INTO** L\_WYCIECZEK **FROM** DOSTEPNE\_WYCIECZKI\_2 **WHERE ID\_WYCIECZKI**=IDWYCIECZKI;  
**IF** L\_WYCIECZEK = 0 **THEN***raise\_application\_error* (-20001,**'PODANA WYCIECZKA JEST NIEDOSTĘPNA'**);  
**END IF**;  
**END**;

Modyfikacja procedury dodającej rezerwacje, poprzez użycie zmodyfikowanych procedur sprawdzających oraz wywołanie procedury PRZELICZ() , powodującej update nowo dodanej kolumny.

**CREATE OR REPLACE PROCEDURE** *DODAJ\_REZERWACJE\_2* (IDWYCIECZKI **NUMBER**, IDOSOBY **NUMBER**) **AS** data\_wycieczki **DATE**;  
 id\_nowej\_rezerwacji **NUMBER**;  
**BEGIN  
 BEGIN** *CZY\_OSOBA\_ISTNIEJE*(IDOSOBY);  
 *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI);  
 *LICZBA\_WOLNYCH\_MIEJSC\_2*(IDWYCIECZKI); *--modyfikacja* **END**;  
  
 **SELECT DATA INTO** data\_wycieczki **FROM** WYCIECZKI  
 **WHERE** IDWYCIECZKI=**ID\_WYCIECZKI**;  
 **IF** data\_wycieczki < *SYSDATE* **THEN** *raise\_application\_error* (-20001,**'WYCIECZKA NIE JEST DOSTĘPNA'**);  
 **end if**;  
 **INSERT INTO** REZERWACJE(**ID\_WYCIECZKI**,**ID\_OSOBY**,**STATUS**) **VALUES** (IDWYCIECZKI,IDOSOBY,**'N'**)  
 **RETURNING NR\_REZERWACJI INTO** id\_nowej\_rezerwacji;  
  
 **INSERT INTO** REZERWACJE\_LOG(**NR\_REZERWACJI**,**DATA**,**STATUS**) **VALUES** (id\_nowej\_rezerwacji,data\_wycieczki,**'N'**);  
 *PRZELICZ*(); *--modyfikacja***END**;

REZERWACJE\_LOG



**create OR REPLACE FUNCTION** *DOSTEPNE\_WYCIECZKI\_FUNKCJA\_2* (Z\_KRAJ **VARCHAR2**,OD\_DATA **DATE**, DO\_DATA **DATE**)  
**RETURN WYCIECZKI\_TABELA IS** licznik **PLS\_INTEGER** :=1;  
 pw **WYCIECZKI\_TABELA**:= **WYCIECZKI\_TABELA**();  
 **CURSOR** CUR\_DOSTEPNE\_WYCIECZKI\_FUNKCJA\_2 **IS  
  
 select KRAJ**, **DATA**, **NAZWA**, **LICZBA\_MIEJSC**, **ID\_WYCIECZKI  
 FROM** WYCIECZKI\_MIEJSCA2 *--modyfikacja* **WHERE LICZBA\_WOLNYCH\_MIEJSC**>0 **AND DATA**<=DO\_DATA **AND DATA**>=OD\_DATA **AND KRAJ**=Z\_KRAJ;  
  
 **BEGIN  
 FOR** i **IN** CUR\_DOSTEPNE\_WYCIECZKI\_FUNKCJA\_2 **LOOP** pw.*EXTEND*;  
 pw(licznik) :=  
 **POJEDYNCZA\_WYCIECZKA**(i.**KRAJ**,i.**DATA**,i.**NAZWA**,i.**LICZBA\_MIEJSC**,i.**ID\_WYCIECZKI**);  
 licznik := licznik+1;  
 **END LOOP**;  
 **RETURN** pw;  
 **END**;

Modyfikacja procedury ZMIEN\_LICZBE\_MIEJSC poprzez update kolumny LICZBA\_WOLNYCH\_MIEJSC w tabeli WYCIECZKI.

**create PROCEDURE** *ZMIEN\_LICZBE\_MIEJSC\_2* (IDWYCIECZKI **NUMBER**, NOWA\_LICZBA\_MIEJSC **NUMBER**) **AS** liczba\_zarezerwowanych\_miejsc **number**;  
**BEGIN  
 BEGIN** *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI);  
 **END**;  
 **SELECT liczba\_zajetych\_miejsc INTO** liczba\_zarezerwowanych\_miejsc **FROM** MIEJSCA\_ZAJĘTE **WHERE ID\_WYCIECZKI**=IDWYCIECZKI;  
 **IF** liczba\_zarezerwowanych\_miejsc > NOWA\_LICZBA\_MIEJSC **THEN** *raise\_application\_error* (-20001,**'NIE MOŻNA ZMIEJSZYC LICZBY MIEJSC Z POWODU ILOSCI REZERWACJI'**);  
 **END IF**;  
  
 **UPDATE** WYCIECZKI **SET LICZBA\_MIEJSC** = NOWA\_LICZBA\_MIEJSC **WHERE** IDWYCIECZKI = **ID\_WYCIECZKI**;  
 **UPDATE** WYCIECZKI **SET LICZBA\_WOLNYCH\_MIEJSC**=NOWA\_LICZBA\_MIEJSC-liczba\_zarezerwowanych\_miejsc  
 **WHERE** IDWYCIECZKI=**ID\_WYCIECZKI**; *--modyfikacja***END**;

Modyfikacja procedury poprzez update pola LICZBA\_WOLNYCH\_MIEJSC w przypadku anulowania rezerwacji.

**create OR REPLACE PROCEDURE** *ZMIEN\_STATUS\_REZERWACJI\_2* (NRREZERWACJI **NUMBER**, NOWY\_STATUS **CHAR**) **AS** stary\_status **CHAR**;  
 idwycieczki **NUMBER**;  
  
**BEGIN  
 BEGIN** *CZY\_REZERWACJA\_ISTNIEJE*(NRREZERWACJI);  
 **END**;  
 **SELECT STATUS INTO** stary\_status **FROM** REZERWACJE **WHERE** NRREZERWACJI=**NR\_REZERWACJI**;  
 **IF** stary\_status =**'A' THEN** *raise\_application\_error* (-20001,**'REZERWACJA ZOSTALA ANULOWANA'**);  
 **ELSIF** stary\_status=NOWY\_STATUS **THEN** *raise\_application\_error* (-20001,**'STATUS REZERWACJI JEST TAKI SAM'**);  
 **ELSIF** stary\_status=**'Z' AND** (NOWY\_STATUS=**'N' OR** NOWY\_STATUS=**'P'**) **THEN** *raise\_application\_error* (-20001,**'REZERWACJA JEST ZAPŁACONA I POTWIERDZONA, NIE MOŻNA WPROWADZIC TAKIEJ ZMIANY'**);  
 **END IF**;  
  
 **SELECT ID\_WYCIECZKI INTO** idwycieczki **FROM** REZERWACJE **WHERE** NRREZERWACJI=**NR\_REZERWACJI**; *--modyfikacja* **UPDATE** REZERWACJE **SET STATUS** = NOWY\_STATUS **WHERE NR\_REZERWACJI** = NRREZERWACJI;  
 **INSERT INTO** REZERWACJE\_LOG(**NR\_REZERWACJI**,**DATA**,**STATUS**) **VALUES** (NRREZERWACJI,*SYSDATE*,NOWY\_STATUS);  
  
 **IF** NOWY\_STATUS **LIKE 'A' THEN** *--modyfikacja* **UPDATE** WYCIECZKI  
 **SET LICZBA\_WOLNYCH\_MIEJSC** = **LICZBA\_WOLNYCH\_MIEJSC** + 1  
 **WHERE ID\_WYCIECZKI** = idwycieczki;  
 **END IF**;  
  
**END**;

# Zadanie 8

## Dodanie rezerwacji i zmiana statusu

**CREATE OR REPLACE TRIGGER** dodanie\_rezerwacji  
**AFTER INSERT OR UPDATE  
ON** REZERWACJE  
**FOR EACH ROW  
BEGIN  
 INSERT INTO** REZERWACJE\_LOG  
 (**NR\_REZERWACJI**, **DATA**, **STATUS**)  
 **VALUES** (:**NEW**.**NR\_REZERWACJI**, *SYSDATE*, :**NEW**.**STATUS**);  
**END**;

## Usuwanie rezerwacji

**CREATE OR REPLACE TRIGGER** usuniecie\_rezerwacji  
**BEFORE DELETE  
ON** REZERWACJE  
**BEGIN** *raise\_application\_error* (-20001,**'NIE MOŻNA USUNĄĆ REZERWACJI'**);  
**END**;

## 

## Uaktualnianie procedur

Polega na usunięciu fragment kodu odpowiedzialnego za wprowadzanie nowych danych do tabeli REZERWACJE\_LOG.

**create PROCEDURE** *DODAJ\_REZERWACJE\_3* (IDWYCIECZKI **NUMBER**, IDOSOBY **NUMBER**) **AS** data\_wycieczki **DATE**;  
 id\_nowej\_rezerwacji **NUMBER**;  
**BEGIN  
 BEGIN** *CZY\_OSOBA\_ISTNIEJE*(IDOSOBY);  
 *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI);  
 *LICZBA\_WOLNYCH\_MIEJSC\_2*(IDWYCIECZKI); *--modyfikacja* **END**;  
  
 **SELECT DATA INTO** data\_wycieczki **FROM** WYCIECZKI  
 **WHERE** IDWYCIECZKI=**ID\_WYCIECZKI**;  
 **IF** data\_wycieczki < *SYSDATE* **THEN** *raise\_application\_error* (-20001,**'WYCIECZKA NIE JEST DOSTĘPNA'**);  
 **end if**;  
 **INSERT INTO** REZERWACJE(**ID\_WYCIECZKI**,**ID\_OSOBY**,**STATUS**) **VALUES** (IDWYCIECZKI,IDOSOBY,**'N'**)  
 **RETURNING NR\_REZERWACJI INTO** id\_nowej\_rezerwacji;  
  
 *--INSERT INTO REZERWACJE\_LOG(NR\_REZERWACJI,DATA,STATUS) VALUES (id\_nowej\_rezerwacji,data\_wycieczki,'N');  
 PRZELICZ*(); *--modyfikacja***END**;

**create PROCEDURE** *ZMIEN\_STATUS\_REZERWACJI\_3* (NRREZERWACJI **NUMBER**, NOWY\_STATUS **CHAR**) **AS** stary\_status **CHAR**;  
  
**BEGIN  
 BEGIN** *CZY\_REZERWACJA\_ISTNIEJE*(NRREZERWACJI);  
 **END**;  
 **SELECT STATUS INTO** stary\_status **FROM** REZERWACJE **WHERE** NRREZERWACJI=**NR\_REZERWACJI**;  
 **IF** stary\_status =**'A' THEN** *raise\_application\_error* (-20001,**'REZERWACJA ZOSTALA ANULOWANA'**);  
 **ELSIF** stary\_status=NOWY\_STATUS **THEN** *raise\_application\_error* (-20001,**'STATUS REZERWACJI JEST TAKI SAM'**);  
 **ELSIF** stary\_status=**'Z' AND** (NOWY\_STATUS=**'N' OR** NOWY\_STATUS=**'P'**) **THEN** *raise\_application\_error* (-20001,**'REZERWACJA JEST ZAPŁACONA I POTWIERDZONA, NIE MOŻNA WPROWADZIC TAKIEJ ZMIANY'**);  
 **END IF**;  
  
 **UPDATE** REZERWACJE **SET STATUS** = NOWY\_STATUS **WHERE NR\_REZERWACJI** = NRREZERWACJI;  
 *--INSERT INTO REZERWACJE\_LOG(NR\_REZERWACJI,DATA,STATUS) VALUES (NRREZERWACJI,SYSDATE,NOWY\_STATUS);***END**;

# Zadanie 9

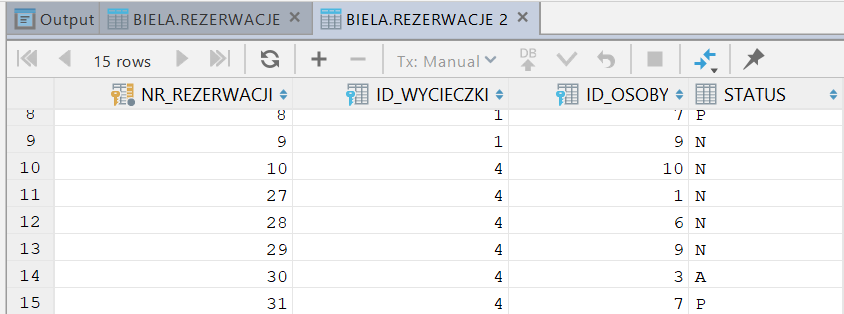
Dodanie wywołania procedury PRZELICZ(), aktualizującej pole LICZBA\_WOLNYCH\_MIEJSC

## Dodanie rezerwacji i zmiana statusu

**CREATE OR REPLACE TRIGGER** dodanie\_rezerwacji  
**AFTER INSERT OR UPDATE  
ON** REZERWACJE  
**FOR EACH ROW  
BEGIN  
 INSERT INTO** REZERWACJE\_LOG(**NR\_REZERWACJI**, **DATA**, **STATUS**)  
 **VALUES** (:**NEW**.**NR\_REZERWACJI**, *SYSDATE*, :**NEW**.**STATUS**);  
  
 **UPDATE** WYCIECZKI **SET LICZBA\_WOLNYCH\_MIEJSC** = **LICZBA\_WOLNYCH\_MIEJSC** - 1  
 **WHERE ID\_WYCIECZKI** = :**NEW**.**ID\_WYCIECZKI**;  
**END**;

**CREATE OR REPLACE TRIGGER** zmienianie\_statusu\_rezerwacji  
**BEFORE INSERT OR UPDATE  
ON** REZERWACJE  
**FOR EACH ROW  
BEGIN  
 INSERT INTO** REZERWACJE\_LOG  
 (**NR\_REZERWACJI**, **DATA**, **STATUS**)  
 **VALUES** (:**NEW**.**NR\_REZERWACJI**, *SYSDATE*, :**NEW**.**STATUS**);  
  
 **IF** :**NEW**.**STATUS** =**'A' THEN  
 UPDATE** WYCIECZKI **SET LICZBA\_WOLNYCH\_MIEJSC** = **LICZBA\_WOLNYCH\_MIEJSC** + 1  
 **WHERE ID\_WYCIECZKI** = :**NEW**.**ID\_WYCIECZKI**;  
 **END IF**;  
**END**;

Zmiana statusu rezerwacji numer 31 z ‘N’ na ‘P’ po wywołaniu funkcji ZMIEN\_STATUS\_REZERWACJI\_3



**create or replace PROCEDURE** *DODAJ\_REZERWACJE\_3* (IDWYCIECZKI **NUMBER**, IDOSOBY **NUMBER**) **AS** data\_wycieczki **DATE**;  
 id\_nowej\_rezerwacji **NUMBER**;  
**BEGIN  
 BEGIN** *CZY\_OSOBA\_ISTNIEJE*(IDOSOBY);  
 *CZY\_WYCIECZKA\_ISTNIEJE*(IDWYCIECZKI);  
 *LICZBA\_WOLNYCH\_MIEJSC*(IDWYCIECZKI);  
 **END**;  
  
 **SELECT DATA INTO** data\_wycieczki **FROM** WYCIECZKI  
 **WHERE** IDWYCIECZKI=**ID\_WYCIECZKI**;  
 **IF** data\_wycieczki < *SYSDATE* **THEN** *raise\_application\_error* (-20001,**'WYCIECZKA NIE JEST DOSTĘPNA'**);  
 **end if**;  
 **INSERT INTO** REZERWACJE(**ID\_WYCIECZKI**,**ID\_OSOBY**,**STATUS**) **VALUES** (IDWYCIECZKI,IDOSOBY,**'N'**)  
 **RETURNING NR\_REZERWACJI INTO** id\_nowej\_rezerwacji;  
  
 *--INSERT INTO REZERWACJE\_LOG(NR\_REZERWACJI,DATA,STATUS) VALUES (id\_nowej\_rezerwacji,data\_wycieczki,'N');***END**;

**create PROCEDURE** *ZMIEN\_STATUS\_REZERWACJI\_3* (NRREZERWACJI **NUMBER**, NOWY\_STATUS **CHAR**) **AS** stary\_status **CHAR**;  
  
**BEGIN  
 BEGIN** *CZY\_REZERWACJA\_ISTNIEJE*(NRREZERWACJI);  
 **END**;  
 **SELECT STATUS INTO** stary\_status **FROM** REZERWACJE **WHERE** NRREZERWACJI=**NR\_REZERWACJI**;  
 **IF** stary\_status =**'A' THEN** *raise\_application\_error* (-20001,**'REZERWACJA ZOSTALA ANULOWANA'**);  
 **ELSIF** stary\_status=NOWY\_STATUS **THEN** *raise\_application\_error* (-20001,**'STATUS REZERWACJI JEST TAKI SAM'**);  
 **ELSIF** stary\_status=**'Z' AND** (NOWY\_STATUS=**'N' OR** NOWY\_STATUS=**'P'**) **THEN** *raise\_application\_error* (-20001,**'REZERWACJA JEST ZAPŁACONA I POTWIERDZONA, NIE MOŻNA WPROWADZIC TAKIEJ ZMIANY'**);  
 **END IF**;  
  
 **UPDATE** REZERWACJE **SET STATUS** = NOWY\_STATUS **WHERE NR\_REZERWACJI** = NRREZERWACJI;  
 *--INSERT INTO REZERWACJE\_LOG(NR\_REZERWACJI,DATA,STATUS) VALUES (NRREZERWACJI,SYSDATE,NOWY\_STATUS);***END**;