Wykonał Wiktor Łęczyński

2.Generator

**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Adam', 'Kowalski', '32354321', 'tel: 6623');  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Jan', 'Nowak', '42145678', 'tel: 2312, dzwonić po 18.00');  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Adam', 'Nowak', '35654321', 'tel: 66232253');  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Patryk', 'Kurzeja', '52345678', 'tel: 4252122');  
  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Wiktor', 'Leczynski', '85454321', 'tel: 52434');  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Tomek', 'Wozniak', '15445678', 'tel: 8653344');  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Adam', 'Czapinski', '95454321', 'tel: 353225');  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Witold', 'Nowakowski', '42345678', 'tel:4242444');  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Karol', 'Peronczyk', '18654321', 'tel: 4221211');  
  
**INSERT INTO** osoby (imie, nazwisko, pesel, kontakt)  
**VALUES**('Agnieszka', 'Nowakowska', '86645678', 'tel: 43123378');  
  
**INSERT INTO** WYCIECZKI (nazwa, kraj, **data**, opis, liczba\_miejsc)  
**VALUES** ('Piękny Kraków','Polska',*TO\_DATE*('2017-02-03','YYYY-MM-DD'),  
 'Najciekawa wycieczka ...',2);  
  
**INSERT INTO** WYCIECZKI (nazwa, kraj, **data**, opis, liczba\_miejsc)  
**VALUES** ('Wieliczka','Polska',*TO\_DATE*('2017-03-03','YYYY-MM-DD'),  
 'Zadziwiająca kopalnia ...',3);  
  
**INSERT INTO** WYCIECZKI (nazwa, kraj, **data**, opis, liczba\_miejsc)  
**VALUES** (' Warszawa','Polska',*TO\_DATE*('2019-02-03','YYYY-MM-DD'),  
 'Najciekawa stolica ...',5);  
  
**INSERT INTO** WYCIECZKI (nazwa, kraj, **data**, opis, liczba\_miejsc)  
**VALUES** ('Niemcy','Niemcy',*TO\_DATE*('2015-03-03','YYYY-MM-DD'),  
 'Zadziwiająca kobiety ...',3);  
  
**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** (3,3,'Z');  
**INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status) **VALUES** (4,4,'A');  
**INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status) **VALUES** (1,5,'N');  
**INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status) **VALUES** (2,6,'P');  
**INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status) **VALUES** (3,7,'Z');  
**INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status) **VALUES** (4,8,'A');  
**INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status) **VALUES** (1,9,'N');  
**INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status) **VALUES** (2,10,'P');

3. Widoki

---3.1  
**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;  
  
 ---3.2  
 **CREATE VIEW** wycieczki\_osoby\_potwierdzone  
 **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  
**where** r.STATUS='P';  
  
--3.3  
**CREATE VIEW** wycieczki\_przyszle  
 **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  
**where** w.DATA>*CURRENT\_TIMESTAMP*;  
  
--3.4  
  
**CREATE or REPLACE VIEW** wycieczki\_miejsca1 **AS SELECT** w.ID\_WYCIECZKI, w.NAZWA,w.KRAJ,w.DATA,  
 w.LICZBA\_MIEJSC - (**SELECT** *COUNT* (\*)  
 **from** REZERWACJE r **where** r.ID\_WYCIECZKI = w.ID\_WYCIECZKI)  
 **AS** "wolne miejsca"  
 **from** WYCIECZKI w  
  
  
--3.5  
  
**CREATE or REPLACE VIEW** wycieczki\_miejsca\_dostepnych **AS SELECT** w.ID\_WYCIECZKI, w.NAZWA,w.KRAJ,w.DATA,  
 w.LICZBA\_MIEJSC - (**SELECT** *COUNT* (\*)  
 **from** REZERWACJE r **where** r.ID\_WYCIECZKI = w.ID\_WYCIECZKI)  
 **AS** "wolne miejsca"  
 **from** WYCIECZKI w  
 **WHERE** w.LICZBA\_MIEJSC - (**SELECT** *COUNT* (\*)  
 **from** REZERWACJE r **where** r.ID\_WYCIECZKI = w.ID\_WYCIECZKI)>0

--3.6  
 **CREATE VIEW** rezerwacje\_do\_anulowania  
 **AS SELECT** w.ID\_WYCIECZKI,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** r.STATUS!='A' **and** w.DATA<*CURRENT\_TIMESTAMP*+7;  
  
  
--dodatkowe  
**CREATE VIEW** wycieczki\_osoby\_id  
 **AS SELECT** w.ID\_WYCIECZKI,w.NAZWA,w.KRAJ,w.DATA,o.IMIE,o.NAZWISKO,r.STATUS,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;

4.Procedury pobierające dane

--4.1  
**CREATE TYPE** uczsetnic\_wycieszki\_row **AS OBJECT** (  
 ID\_WYCIECZKI **INT**,  
 NAZWA **VARCHAR2**(100),  
 KRAJ **VARCHAR2**(50),  
 **DATA DATE**,  
 IMIE **VARCHAR2**(50),  
 NAZWISKO **VARCHAR2**(50),  
 STATUS **CHAR**(1)  
)  
**CREATE TYPE** UCZESTNICY\_WYC\_TABLE **AS TABLE OF** UCZSETNIC\_WYCIESZKI\_ROW  
  
  
**CREATE FUNCTION** *uczestnicy\_wycieczki2* (id\_wycieszki **IN INT**)  
 **RETURN** uczestnicy\_wyc\_table **PIPELINED IS** curr **SYS\_REFCURSOR**;  
 result\_row uczsetnic\_wycieszki\_row  
 := uczsetnic\_wycieszki\_row(**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**);  
**BEGIN  
 OPEN** curr **FOR  
 SELECT** \* **FROM** WYCIECZKI\_OSOBY **WHERE** ID\_WYCIECZKI= id\_wycieszki;  
 **LOOP  
 FETCH** curr **INTO** result\_row.ID\_WYCIECZKI, result\_row.nazwa, result\_row.KRAJ, result\_row.DATA, result\_row.IMIE,  
 result\_row.NAZWISKO, result\_row.STATUS;  
 **EXIT WHEN** curr%**NOTFOUND**;  
 **PIPE ROW**(result\_row);  
 **END LOOP**;  
 **CLOSE** curr;  
 **RETURN**;  
**END**;  
  
--4.2  
**CREATE TYPE** uczsetnic\_rezerwacja\_row **AS OBJECT** (  
 ID\_WYCIECZKI **INT**,  
 NAZWA **VARCHAR2**(100),  
 KRAJ **VARCHAR2**(50),  
 **DATA DATE**,  
 IMIE **VARCHAR2**(50),  
 NAZWISKO **VARCHAR2**(50),  
 STATUS **CHAR**(1),  
 ID\_OSOBY **INT**)  
/  
-- auto-generated definition  
**CREATE TYPE** UCZESTNICY\_REZ\_TABLE **AS TABLE OF** UCZSETNIC\_REZERWACJA\_ROW  
/  
  
 **CREATE FUNCTION** *rezerwacje\_osob* (id\_osoby1 **IN INT**)  
 **RETURN** uczestnicy\_rez\_table **PIPELINED IS** curr **SYS\_REFCURSOR**;  
 result\_row UCZSETNIC\_REZERWACJA\_ROW  
 := UCZSETNIC\_REZERWACJA\_ROW(**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**);  
**BEGIN  
 OPEN** curr **FOR  
 SELECT** \* **FROM** WYCIECZKI\_OSOBY\_ID **WHERE** ID\_OSOBY= id\_osoby1;  
 **LOOP  
 FETCH** curr **INTO** result\_row.ID\_WYCIECZKI,  
 result\_row.nazwa,  
 result\_row.KRAJ,  
 result\_row.DATA,  
 result\_row.IMIE,  
 result\_row.NAZWISKO,  
 result\_row.STATUS,  
 result\_row.ID\_OSOBY;  
 **EXIT WHEN** curr%**NOTFOUND**;  
 **PIPE ROW**(result\_row);  
 **END LOOP**;  
 **CLOSE** curr;  
 **RETURN**;  
**END**;  
/  
  
--4.3  
  
 **CREATE FUNCTION** *przyszle\_rezerwacje\_osob* (id\_osoby1 **IN INT**)  
 **RETURN** uczestnicy\_rez\_table **PIPELINED IS** curr **SYS\_REFCURSOR**;  
 result\_row UCZSETNIC\_REZERWACJA\_ROW  
 := UCZSETNIC\_REZERWACJA\_ROW(**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**);  
**BEGIN  
 OPEN** curr **FOR  
 SELECT** \* **FROM** WYCIECZKI\_OSOBY\_ID **WHERE** ID\_OSOBY= id\_osoby1  
 **and DATA**>*CURRENT\_TIMESTAMP* ;  
 **LOOP  
 FETCH** curr **INTO** result\_row.ID\_WYCIECZKI,  
 result\_row.nazwa,  
 result\_row.KRAJ,  
 result\_row.DATA,  
 result\_row.IMIE,  
 result\_row.NAZWISKO,  
 result\_row.STATUS,  
 result\_row.ID\_OSOBY;  
 **EXIT WHEN** curr%**NOTFOUND**;  
 **PIPE ROW**(result\_row);  
 **END LOOP**;  
 **CLOSE** curr;  
 **RETURN**;  
**END**;  
--4.4  
  
**CREATE or REPLACE TYPE** wycieczki\_row2 **AS OBJECT** (  
 ID\_WYCIECZKI **INT**,  
 NAZWA **VARCHAR2**(100),  
 KRAJ **VARCHAR2**(50),  
 **DATA DATE**,  
 liczba\_miejsc **INT**,  
 liczba\_wolnych\_miejsc **int**)  
/

-- auto-generated definition  
**CREATE or REPLACE TYPE** WYCIECZKI\_TABLE **AS TABLE OF** wycieczki\_row2  
  
/  
 **CREATE or REPLACE FUNCTION** *dostepne\_wycieczki2* (kraj **in VARCHAR2**,  
 data\_od **in DATE**, data\_do **in DATE**)  
 **RETURN** WYCIECZKI\_TABLE **PIPELINED IS** curr **SYS\_REFCURSOR**;  
 result\_row wycieczki\_row2  
 := wycieczki\_row2(**NULL**,**NULL**,**NULL**,**NULL**,**NULL**,**NULL**);  
**BEGIN  
 OPEN** curr **FOR  
 SELECT** \* **FROM** WYCIECZKI\_MIEJSCA\_DOSTEPNYCH  
 **WHERE** kraj=WYCIECZKI\_MIEJSCA\_DOSTEPNYCH.KRAJ  
 **and** data\_od<WYCIECZKI\_MIEJSCA\_DOSTEPNYCH.DATA  
 **and** data\_do>WYCIECZKI\_MIEJSCA\_DOSTEPNYCH.DATA ;  
 **LOOP  
 FETCH** curr **INTO** result\_row.ID\_WYCIECZKI,  
 result\_row.nazwa,  
 result\_row.KRAJ,  
 result\_row.DATA,  
 result\_row.liczba\_miejsc,  
 result\_row.liczba\_wolnych\_miejsc;  
 **EXIT WHEN** curr%**NOTFOUND**;  
 **PIPE ROW**(result\_row);  
 **END LOOP**;  
 **CLOSE** curr;  
 **RETURN**;  
**END**;

5.Procedury modyfikujących dane

---5.1  
 **CREATE or REPLACE PROCEDURE** *wstaw\_rezerwacje1* (id\_wycieczki1 **in INT** ,  
 id\_osoby1 **IN INT**)  
 **AS**osoba **INT**;  
wycieczka **INT**;  
ilosc **INT**;  
limit1 **INT**;  
czas **DATE**;  
  
**BEGIN  
 BEGIN  
 SELECT** *count*(osoby.id\_osoby) **into** osoba **FROM** osoby  
 **where** OSOBY.ID\_OSOBY = id\_osoby1;  
  
 **SELECT** *count*(wycieczki.id\_wycieczki) **into** wycieczka **from** WYCIECZKI  
 **where** WYCIECZKI.ID\_WYCIECZKI = id\_wycieczki1;  
  
 **select** *count*(\*) **into** ilosc **from** rezerwacje  
 **where** REZERWACJE.id\_wycieczki = id\_wycieczki1  
 **and** REZERWACJE.STATUS != 'A';  
  
 **select** *count*(\*) **into** ilosc  
 **from** rezerwacje  
 **where** REZERWACJE.id\_wycieczki = id\_wycieczki1  
 **and** REZERWACJE.STATUS != 'A';  
 **SELECT DATA into** czas **from** WYCIECZKI  
 **WHERE** id\_wycieczki1=WYCIECZKI.ID\_WYCIECZKI;  
  
 **select** LICZBA\_MIEJSC **into** limit1  
 **from** WYCIECZKI **where** id\_wycieczki1=ID\_WYCIECZKI;  
 **END**;  
  
 **if**(osoba = 0)  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma osoby o takim id ' , **TRUE**);  
 **END if**;  
  
**if**(wycieczka = 0 )  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma wycieczki o takim id' , **TRUE**);  
 **end if**;  
  
  
 **if**(ilosc + 1 <= limit1)  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma miejsc' , **TRUE**);  
 **END IF**;  
  
 **if**(*CURRENT\_TIMESTAMP*>czas) **then** *raise\_application\_error* (-20001, 'Wycieczka juz sie odbyla' , **TRUE**);  
 **END IF**;  
  
 **INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status)  
**VALUES** (id\_wycieczki1,id\_osoby1,'N');  
  
  
**END** *wstaw\_rezerwacje1*;

---5.2  
**CREATE or REPLACE PROCEDURE** *zmien\_status\_rezerwacji* (id\_rezerwacji1 **in INT** ,  
 status1 **IN CHAR**)  
 **AS** rezerwacja **INT**;  
**BEGIN  
 BEGIN  
SELECT** *count*(REZERWACJE.NR\_REZERWACJI) **into** rezerwacja **from** REZERWACJE  
 **where** NR\_REZERWACJI=id\_rezerwacji1;  
 **END**;  
 **if**(rezerwacja = 0 )  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma wycieczki o takim id' , **TRUE**);  
 **end if**;  
  
 **UPDATE** REZERWACJE **set** STATUS=status1  
 **where** id\_rezerwacji1=NR\_REZERWACJI;  
**END** ;

--5.3  
  
**CREATE or REPLACE PROCEDURE** *zmien\_liczbe\_miejsc* (id\_wycieczki **in INT** ,  
 liczba\_miejsc **IN INT**)  
 **AS** wycieczka **INT**;  
 liczba **INT**;  
**BEGIN  
 BEGIN  
SELECT** *count*(WYCIECZKI.ID\_WYCIECZKI) **into** wycieczka **from** WYCIECZKI  
 **where** WYCIECZKI.ID\_WYCIECZKI=id\_wycieczki;  
  
**SELECT** *count*(\*) **into** liczba **from** WYCIECZKI w  
**INNER JOIN** REZERWACJE r **on** r.ID\_WYCIECZKI=w.ID\_WYCIECZKI  
**where** w.ID\_WYCIECZKI=id\_wycieczki **and** r.STATUS!='A';  
  
 **END**;  
 **if**(wycieczka = 0 )  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma wycieczki o takim id' , **TRUE**);  
 **end if**;  
**if**(liczba>liczba\_miejsc)  
 **THEN** *raise\_application\_error* (-20001, 'Za duzo rejestracji' , **TRUE**);  
**END IF**;  
  
 **UPDATE** WYCIECZKI **set** LICZBA\_MIEJSC=liczba\_miejsc  
 **where** ID\_WYCIECZKI=id\_wycieczki;  
**END** ;

6.Dziennik

---6.1  
 **CREATE or REPLACE PROCEDURE** *wstaw\_rezerwacje1\_log* (id\_wycieczki1 **in INT** ,  
 id\_osoby1 **IN INT**)  
 **AS**osoba **INT**;  
wycieczka **INT**;  
ilosc **INT**;  
limit1 **INT**;  
czas **DATE**;  
  
**BEGIN  
 BEGIN  
 SELECT** *count*(osoby.id\_osoby) **into** osoba **FROM** osoby  
 **where** OSOBY.ID\_OSOBY = id\_osoby1;  
  
 **SELECT** *count*(wycieczki.id\_wycieczki) **into** wycieczka **from** WYCIECZKI  
 **where** WYCIECZKI.ID\_WYCIECZKI = id\_wycieczki1;  
  
 **select** *count*(\*) **into** ilosc **from** rezerwacje  
 **where** REZERWACJE.id\_wycieczki = id\_wycieczki1  
 **and** REZERWACJE.STATUS != 'A';  
  
 **select** *count*(\*) **into** ilosc  
 **from** rezerwacje  
 **where** REZERWACJE.id\_wycieczki = id\_wycieczki1  
 **and** REZERWACJE.STATUS != 'A';  
 **SELECT DATA into** czas **from** WYCIECZKI  
 **WHERE** id\_wycieczki1=WYCIECZKI.ID\_WYCIECZKI;  
  
 **select** LICZBA\_MIEJSC **into** limit1  
 **from** WYCIECZKI **where** id\_wycieczki1=ID\_WYCIECZKI;  
 **END**;  
  
 **if**(osoba = 0)  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma osoby o takim id ' , **TRUE**);  
 **END if**;  
  
**if**(wycieczka = 0 )  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma wycieczki o takim id' , **TRUE**);  
 **end if**;  
  
  
 **if**(ilosc + 1 <= limit1)  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma miejsc' , **TRUE**);  
 **END IF**;  
  
 **if**(*CURRENT\_TIMESTAMP*>czas) **then** *raise\_application\_error* (-20001, 'Wycieczka juz sie odbyla' , **TRUE**);  
 **END IF**;  
  
 **INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status)  
**VALUES** (id\_wycieczki1,id\_osoby1,'N');  
  
  
 **BEGIN  
 SELECT** *MAX*(REZERWACJE.NR\_REZERWACJI) **into** rezerwacja **from** REZERWACJE  
 **order BY** REZERWACJE.NR\_REZERWACJI **DESC**;  
**END**;  
  
 **INSERT INTO** REZERWACJE\_LOG(ID\_REZERWACJI,**DATA**, STATUS)  
 **VALUES** (rezerwacja,*current\_timestamp*,'N');  
  
  
**END** *wstaw\_rezerwacje1\_log*;

--6.2

**CREATE or REPLACE PROCEDURE** *zmien\_status\_rezerwacji\_log* (id\_rezerwacji1 **in INT** ,  
 status1 **IN CHAR**)  
 **AS** rezerwacja **INT**;  
**BEGIN  
 BEGIN  
SELECT** *count*(REZERWACJE.NR\_REZERWACJI) **into** rezerwacja **from** REZERWACJE  
 **where** NR\_REZERWACJI=id\_rezerwacji1;  
 **END**;  
 **if**(rezerwacja = 0 )  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma wycieczki o takim id' , **TRUE**);  
 **end if**;  
  
 **UPDATE** REZERWACJE **set** STATUS=status1  
 **where** id\_rezerwacji1=NR\_REZERWACJI;  
  
  
  
 **INSERT INTO** REZERWACJE\_LOG(ID\_REZERWACJI,**DATA**, STATUS)  
 **VALUES** (id\_rezerwacji1,*current\_timestamp*,'U');  
  
**END** ;

7.Wolne\_miejsca

**ALTER TABLE** WYCIECZKI **ADD** liczba\_wolnych\_miejsc **INT**;  
  
--7.1  
**CREATE VIEW** wycieczki\_miejsca\_w  
 **AS SELECT** w.ID\_WYCIECZKI,w.NAZWA,w.KRAJ,w.DATA,w.LICZBA\_WOLNYCH\_MIEJSC  
 **FROM** WYCIECZKI w;  
--7.2  
**CREATE VIEW** wycieczki\_miejsca\_dostepnych  
 **AS SELECT** w.ID\_WYCIECZKI,w.NAZWA,w.KRAJ,w.DATA,w.LICZBA\_WOLNYCH\_MIEJSC  
 **FROM** WYCIECZKI w  
**where** LICZBA\_WOLNYCH\_MIEJSC>0;  
--7.3  
  
**CREATE or REPLACE PROCEDURE** *Wylicz\_wolne\_miejsca***AS  
  
BEGIN  
 for** v\_wyc **IN** (**SELECT** ID\_WYCIECZKI,LICZBA\_MIEJSC,LICZBA\_WOLNYCH\_MIEJSC **FROM** WYCIECZKI) **LOOP  
  
 Update** WYCIECZKI **set** LICZBA\_WOLNYCH\_MIEJSC=v\_wyc.LICZBA\_MIEJSC-(**SELECT** *count*(\*) **FROM** REZERWACJE  
 **where** STATUS!='A' **and** REZERWACJE.ID\_WYCIECZKI=v\_wyc.ID\_WYCIECZKI)  
 **WHERE** WYCIECZKI.ID\_WYCIECZKI=v\_wyc.ID\_WYCIECZKI;  
  
 **END LOOP**;  
  
**END** *Wylicz\_wolne\_miejsca*;

--7.4

---5.1  
 **CREATE or REPLACE PROCEDURE** *wstaw\_rezerwacje1* (id\_wycieczki1 **in INT** ,  
 id\_osoby1 **IN INT**)  
 **AS**osoba **INT**;  
wycieczka **INT**;  
ilosc **INT**;  
czas **DATE**;  
  
**BEGIN  
 BEGIN  
 SELECT** *count*(osoby.id\_osoby) **into** osoba **FROM** osoby  
 **where** OSOBY.ID\_OSOBY = id\_osoby1;  
  
 **SELECT** *count*(wycieczki.id\_wycieczki) **into** wycieczka **from** WYCIECZKI  
 **where** WYCIECZKI.ID\_WYCIECZKI = id\_wycieczki1;  
  
 **select** *count*(\*) **into** ilosc **from** rezerwacje  
 **where** REZERWACJE.id\_wycieczki = id\_wycieczki1  
 **and** REZERWACJE.STATUS != 'A';  
  
 **select** WYCIECZKI.LICZBA\_WOLNYCH\_MIEJSC **into** ilosc  
 **from** WYCIECZKI  
 **where** id\_wycieczki1=ID\_WYCIECZKI;  
  
 **SELECT DATA into** czas **from** WYCIECZKI  
 **WHERE** id\_wycieczki1=WYCIECZKI.ID\_WYCIECZKI;  
  
  
  
 **if**(osoba = 0)  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma osoby o takim id ' , **TRUE**);  
 **END if**;  
  
**if**(wycieczka = 0 )  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma wycieczki o takim id' , **TRUE**);  
 **end if**;  
  
  
 **if**(ilosc <0)  
 **THEN** *raise\_application\_error* (-20001, 'Nie ma miejsc' , **TRUE**);  
 **END IF**;  
  
 **if**(*CURRENT\_TIMESTAMP*>czas) **then** *raise\_application\_error* (-20001, 'Wycieczka juz sie odbyla' , **TRUE**);  
 **END IF**;  
  
 **INSERT INTO** rezerwacje(id\_wycieczki, id\_osoby, status)  
**VALUES** (id\_wycieczki1,id\_osoby1,'N');  
  
  
**END** *wstaw\_rezerwacje1*;

8.Trigger\_dziennik

**CREATE or REPLACE TRIGGER** Dziennik  
 **AFTER INSERT or UPDATE or DELETE ON** REZERWACJE  
 **FOR EACH ROW  
 BEGIN  
if INSERTING then  
 INSERT INTO** REZERWACJE\_LOG(ID\_REZERWACJI,**DATA**, STATUS)  
 **VALUES** ( :**new**.NR\_REZERWACJI,*current\_timestamp*,'R');  
 **END IF**;  
  
 **if UPDATING then  
 INSERT INTO** REZERWACJE\_LOG(ID\_REZERWACJI,**DATA**, STATUS)  
 **VALUES** ( :**new**.NR\_REZERWACJI,*current\_timestamp*,'U');  
 **END IF**;  
  
 **if DELETING then  
 INSERT INTO** REZERWACJE\_LOG(ID\_REZERWACJI,**DATA**, STATUS)  
 **VALUES** ( :**new**.NR\_REZERWACJI,*current\_timestamp*,'D');  
 **END IF**;  
  
 **END**;

9.Trigger\_wolnemiejsca

**create or replace TRIGGER** zaktualizuj\_wolne\_miejsca  
 **AFTER INSERT or UPDATE or DELETE on** REZERWACJE **for EACH ROW  
 BEGIN  
 if INSERTING then  
 update** WYCIECZKI  
 **set** LICZBA\_WOLNYCH\_MIEJSC = LICZBA\_WOLNYCH\_MIEJSC -1  
 **where** WYCIECZKI.ID\_WYCIECZKI = :**NEW**.ID\_WYCIECZKI;  
 **ELSEIF UPDATING then  
 IF** :**NEW**.STATUS='A' **THEN  
 update** WYCIECZKI  
 **set** LICZBA\_WOLNYCH\_MIEJSC = LICZBA\_WOLNYCH\_MIEJSC +1  
 **where** WYCIECZKI.ID\_WYCIECZKI = :**NEW**.ID\_WYCIECZKI;  
 **END IF**;  
 **IF** :**OLD**.STATUS='A' **THEN  
 update** WYCIECZKI  
 **set** LICZBA\_WOLNYCH\_MIEJSC = LICZBA\_WOLNYCH\_MIEJSC -1  
 **where** WYCIECZKI.ID\_WYCIECZKI = :**NEW**.ID\_WYCIECZKI;  
 **END IF**;  
 **ELSE  
 update** WYCIECZKI  
 **set** LICZBA\_WOLNYCH\_MIEJSC = LICZBA\_WOLNYCH\_MIEJSC +1  
 **where** WYCIECZKI.ID\_WYCIECZKI = :**NEW**.ID\_WYCIECZKI;  
 **END IF** ;  
 **END**;