Marek Fudaliński

Sprawozdanie Laboratorium 1 – Bazy Danych

1. Tabele

Stworzenie tabeli 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**);

Stworzenie tabeli 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**);

Dodanie tablei 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**);

Dodanie relacji pomiędzy tabelą rezerwacje a osoby

**ALTER TABLE** REZERWACJE  
**ADD CONSTRAINT** REZERWACJE\_FK1 **FOREIGN KEY**(  
 **ID\_OSOBY**)  
**REFERENCES** OSOBY  
(  
 **ID\_OSOBY**)  
**ENABLE**;

Dodanie relacji między tabela rezerwacje a wycieczki

**ALTER TABLE** REZERWACJE  
**ADD CONSTRAINT** REZERWACJE\_FK2 **FOREIGN KEY**(  
 **ID\_WYCIECZKI**)  
**REFERENCES** WYCIECZKI  
(  
 **ID\_WYCIECZKI**)  
**ENABLE**;

Nałożenie check’a na kolumnę status w tabeli rezerwacje

**ALTER TABLE** REZERWACJE  
**ADD CONSTRAINT** REZERWACJE\_CHK1 **CHECK**(**status IN** (**'N'**,**'P'**,**'Z'**,**'A'**))  
**ENABLE**;

2. Wypełniamy tabele przykładowymi danymi danymi

4 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** (**'Gniezno'**,**'Polska'**,**'2019-03-03'**,**'Zadziwiająca miasto ...'**,2);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Moskwa'**,**'Rosja'**,**'2018-03-03'**,**'Zadziwiająca miasto ...'**,4);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Wilno'**,**'Litwa'**,**'2016-03-03'**,**'Zadziwiająca miasto ...'**,7);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Mińsk'**,**'Białoruś'**,**'2019-06-03'**,**'Zadziwiająca miasto ...'**,22);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Kijów'**,**'Ukraina'**,**'2018-12-03'**,**'Zadziwiająca miasto ...'**,12);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Bratysława'**,**'Słowacja'**,**'2018-11-23'**,**'Zadziwiająca miasto ...'**,32);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Praga'**,**'Czechy'**,**'2020-10-03'**,**'Zadziwiająca miasto ...'**,10);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Monachium'**,**'Niemcy'**,**'2014-03-03'**,**'Zadziwiająca miasto ...'**,18);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Wieliczka2'**,**'Polska'**,*TO\_DATE*(**'2017-03-03'**,**'YYYY-MM-DD'**),  
 **'Zadziwiająca kopalnia ...'**,2);  
**INSERT INTO** wycieczki (**nazwa**, **kraj**, **data**, **opis**, **liczba\_miejsc**)  
**VALUES** (**'Wieliczka23'**,**'Polska'**,*TO\_DATE*(**'2019-03-03'**,**'YYYY-MM-DD'**),  
 **'Zadziwiająca kopalnia myslenicka w wieliczce...'**,4);

10 osób

**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Adam'**, **'Kowalski'**, **'87654321'**, **'tel: 6623'**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Jan'**, **'Nowak'**, **'12345678'**, **'tel: 231322, nie dzwonić po 18.00'**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Ala'**, **'Kowalska'**, **'12345678'**, **'tel: 245312,nie dzwonić'**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Stansiław'**, **'Nowak'**, **'12345678'**, **'tel: 762312,'**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Jakub'**, **'Kowalska'**, **'12345678'**, **'tel: 982312,tak'**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Mirosław'**, **'Nowak'**, **'12345678'**, **'tel: 221312, '**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Bartek'**, **'Kowalska'**, **'12345678'**, **'tel: 254312, domowy'**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Marcin'**, **'Kowalska'**, **'12345678'**, **'tel: 276312, stacjonarny'**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Małgorzata'**, **'Kowalska'**, **'12345678'**, **'tel: 2979312, satelitarny'**);  
**INSERT INTO** osoby (**imie**, **nazwisko**, **pesel**, **kontakt**)  
**VALUES**(**'Marzena'**, **'Kowalska'**, **'12345678'**, **'tel: 2344312, Nokia 6230i'**);

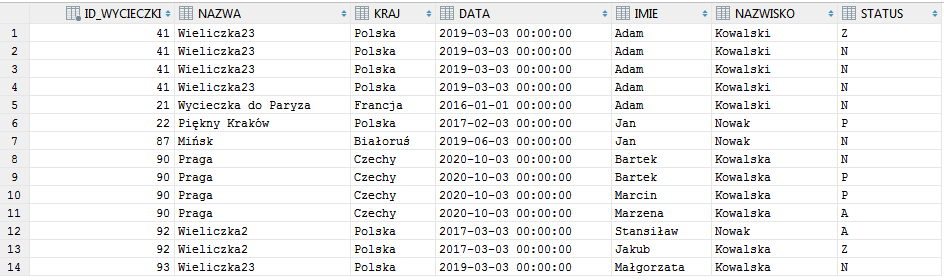
10 rezerwacji

**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (21,21,**'N'**);  
**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (22,22,**'P'**);**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (87,42,**'N'**);  
**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (92,44,**'A'**);  
**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (92,45,**'Z'**);  
**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (90,47,**'N'**);  
**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (93,49,**'N'**);  
**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (90,47,**'P'**);  
**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (90,48,**'P'**);  
**INSERT INTO** rezerwacje(**id\_wycieczki**, **id\_osoby**, **status**)  
**VALUES** (90,50,**'A'**);

3. Tworzenie widoków. Należy przygotować kilka widoków ułatwiających dostęp do danych  
a) wycieczki\_osoby(kraj,data, nazwa\_wycieczki, imie, nazwisko,status\_rezerwacji)

**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**;

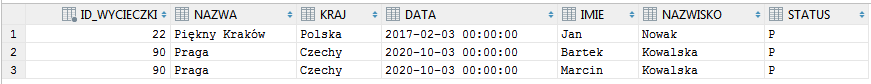
SELECT t.\* FROM MFUDALIN.WYCIECZKI\_OSOBY t



b) wycieczki\_osoby\_potwierdzone (kraj,data, nazwa\_wycieczki, imie, nazwisko,status\_rezerwacji)

**CREATE VIEW** wycieczki\_osoby\_potwierdzone  
 **AS  
 SELECT** r.**ID\_WYCIECZKI**,  
 r.**NAZWA**,  
 r.**KRAJ**,  
 r.**DATA**,  
 r.**IMIE**,  
 r.**NAZWISKO**,  
 r.**STATUS  
 FROM** wycieczki\_osoby r  
 **where** r.**STATUS** = **'P'**;

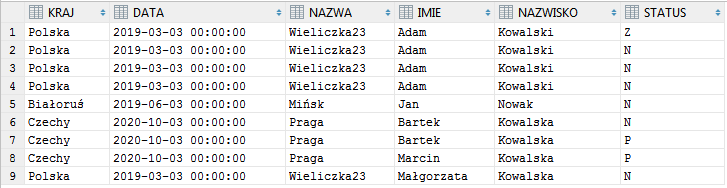
SELECT t.\* FROM MFUDALIN.WYCIECZKI\_OSOBY\_POTWIERDZONE t



c) wycieczki\_przyszle (kraj,data, nazwa\_wycieczki, imie, nazwisko,status\_rezerwacji)

**CREATE or replace VIEW** wycieczki\_przyszle  
 **AS  
 SELECT** w.**KRAJ**,  
 w.**DATA**,  
 w.**NAZWA**,  
 o.**IMIE**,  
 o.**NAZWISKO**,  
 r.**STATUS  
 FROM** wycieczki w  
 **JOIN** REZERWACJE r **on** w.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI and** r.**STATUS** != **'A'  
 JOIN** OSOBY o **on** r.**ID\_OSOBY** = o.**ID\_OSOBY  
 where** w.**DATA** - *SYSDATE* > 0;  
**commit** ;

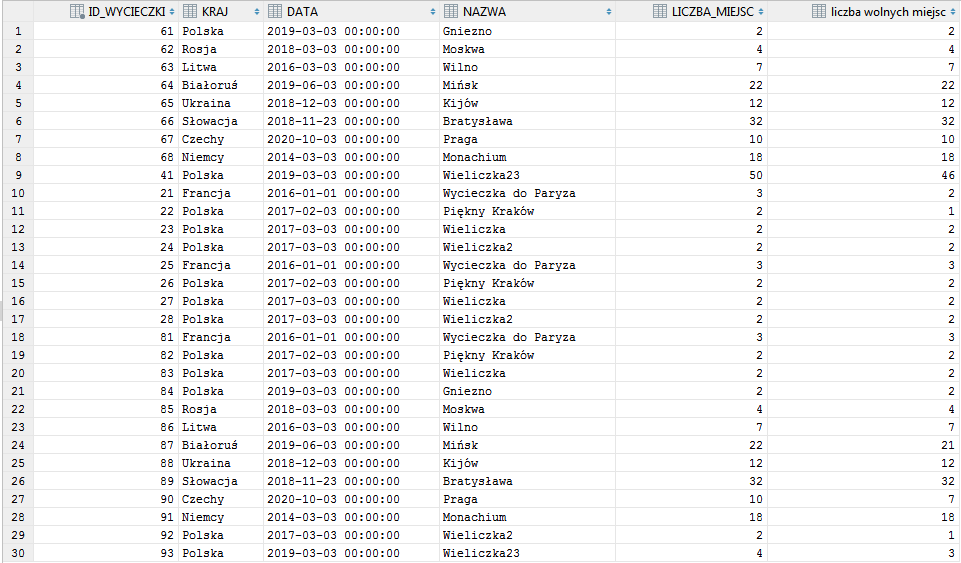
SELECT t.\* FROM MFUDALIN.WYCIECZKI\_PRZYSZLE t



d) wycieczki\_miejsca(kraj,data, nazwa\_wycieczki,liczba\_miejsc, liczba\_wolnych\_miejsc)

**CREATE or replace view** wycieczki\_miejsca  
 **as  
 SELECT** w.**ID\_WYCIECZKI**,  
 w.**KRAJ**,  
 w.**DATA**,  
 w.**NAZWA**,  
 w.**LICZBA\_MIEJSC**,  
 w.**LICZBA\_MIEJSC** - (  
 **select** *count*(r.**ID\_OSOBY**) **FROM** wycieczki w1  
 **JOIN** REZERWACJE r **on** w1.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI  
 JOIN** OSOBY o **on** r.**ID\_OSOBY** = o.**ID\_OSOBY  
 where** r.**STATUS** != **'A' and** w1.**ID\_WYCIECZKI** = w.**ID\_WYCIECZKI**) **as** "liczba wolnych miejsc"  
 **from** wycieczki w;

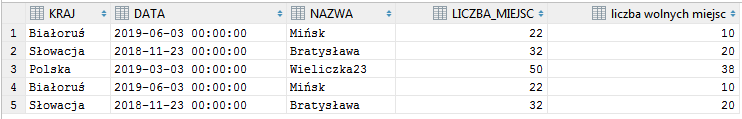
SELECT t.\* FROM MFUDALIN.WYCIECZKI\_MIEJSCA t



e) dostępne\_wyciezki(kraj,data, nazwa\_wycieczki,liczba\_miejsc, liczba\_wolnych\_miejsc)

**CREATE view** dostepne\_wycieczki  
 **as  
 SELECT** w.**KRAJ**,  
 w.**DATA**,  
 w.**NAZWA**,  
 w.**LICZBA\_MIEJSC**,  
 w.**LICZBA\_MIEJSC** - (  
 **select** *count*(o.**ID\_OSOBY**) **FROM** wycieczki w1  
 **JOIN** REZERWACJE r **on** w1.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI  
 JOIN** OSOBY o **on** r.**ID\_OSOBY** = o.**ID\_OSOBY  
 where** r.**STATUS** != **'A'**) **as** "liczba wolnych miejsc"  
 **from** wycieczki w  
 **where** w.**LICZBA\_MIEJSC** - (  
 **select** *count*(o.**ID\_OSOBY**) **FROM** wycieczki w1  
 **JOIN** REZERWACJE r **on** w1.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI  
 JOIN** OSOBY o **on** r.**ID\_OSOBY** = o.**ID\_OSOBY  
 where** r.**STATUS** != **'A'** ) > 0  
 **and** w.**DATA** - *SYSDATE* > 0;

SELECT t.\* FROM MFUDALIN.DOSTĘPNE\_WYCIECZKI t



f) rezerwacje\_do\_ anulowania – lista niepotwierdzonych rezerwacji które powinne zostać anulowane, rezerwacje przygotowywane są do anulowania na tydzień przed wyjazdem)

**CREATE VIEW** rezerwacje\_do\_anulowania  
 **as  
 SELECT** r.**ID\_OSOBY**,r.**ID\_WYCIECZKI**,w.**NAZWA from** WYCIECZKI w  
 **join** REZERWACJE r **on** r.**ID\_WYCIECZKI** = w.**ID\_WYCIECZKI and** w.**DATA** - *SYSDATE* < 8  
 **where** r.**STATUS** = **'N'**;

SELECT t.\* FROM MFUDALIN.REZERWACJE\_DO\_ANULOWANIA t

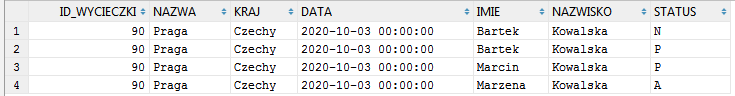


4. Tworzenie procedur/funkcji pobierających dane. Podobnie jak w poprzednim przykładzie należy przygotować kilka procedur ułatwiających dostęp do danych

a) uczestnicy\_wycieczki (id\_wycieczki), procedura ma zwracać podobny zestaw danych jak widok wycieczki\_osoby

**Create or replace function** *uczestnicy\_wycieczki* (fid\_wycieczki **in NUMBER**)  
**return SYS\_REFCURSOR  
IS  
 out SYS\_REFCURSOR**;  
 check\_existance **NUMBER**;  
**BEGIN  
 SELECT** *count*(*\**) **INTO** check\_existance **from** WYCIECZKI w1 **where** fid\_wycieczki = w1.**ID\_WYCIECZKI**;  
 **IF** check\_existance = 0  
 **THEN  
 raise** VALUE\_error;  
 **END IF**;  
 **open out for  
 SELECT** *\** **FROM** wycieczki\_osoby  
 **WHERE ID\_WYCIECZKI**=fid\_wycieczki;  
 **return out**;  
**END**;

**declare  
 result sys\_refcursor**;  
 FID\_WYCIECZKI **NUMBER** := 90;  
**begin  
 result** := *UCZESTNICY\_WYCIECZKI*(FID\_WYCIECZKI => FID\_WYCIECZKI);  
 ? := **result**;  
**end**;

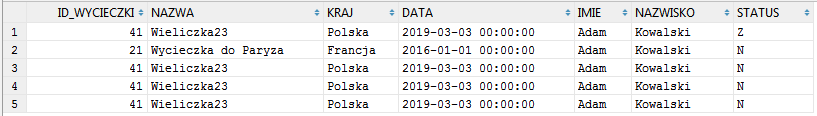


b) rezerwacje\_osoby(id\_osoby), procedura ma zwracać podobny zestaw danych jak widok wycieczki\_osoby

**CREATE OR REPLACE FUNCTION** *rezerwacje\_osoby*(fid\_osoby **in number**)  
 **RETURN SYS\_REFCURSOR  
 IS  
 out sys\_refcursor**;  
 check\_existance **number**;  
 **BEGIN  
 SELECT** *COUNT*(*\**) **INTO** check\_existance **FROM** OSOBY O

**WHERE** O.**ID\_OSOBY** = fid\_osoby;  
 **IF** check\_existance = 0 **THEN  
 RAISE** value\_error ;  
 **end if**;  
 **open out for  
 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** = fid\_osoby;  
 **RETURN out**;  
 **end**;

**declare  
 result sys\_refcursor**;  
 FID\_OSOBY **NUMBER** := 21;  
**begin  
 result** := *REZERWACJE\_OSOBY*(FID\_OSOBY => FID\_OSOBY);  
 ? := **result**;  
**end**;

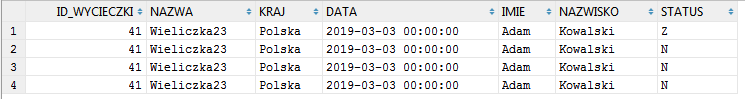


c) przyszle\_rezerwacje\_osoby(id\_osoby)

**CREATE OR REPLACE FUNCTION** *przyszle\_rezerwacje\_osoby*(fid\_osoby **in NUMBER**)  
 **return SYS\_REFCURSOR  
 is** check\_existance **number**;  
 **out sys\_refcursor**;  
 **begin  
 SELECT** *COUNT*(*\**) **INTO** check\_existance **FROM** OSOBY O

**WHERE** O.**ID\_OSOBY** = fid\_osoby;  
 **IF** check\_existance = 0 **THEN  
 RAISE** value\_error ;  
 **end if**;  
 **open out for  
 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** = fid\_osoby **and** w.**DATA** - *sysdate* > 0;  
 **return out**;  
 **end**;

**declare  
 result sys\_refcursor**;  
 FID\_OSOBY **NUMBER** := 21;  
**begin  
 result** := *PRZYSZLE\_REZERWACJE\_OSOBY*(FID\_OSOBY => FID\_OSOBY);  
 ? := **result**;  
**end**;



d) dostepne\_wycieczki(kraj, data\_od, data\_do)

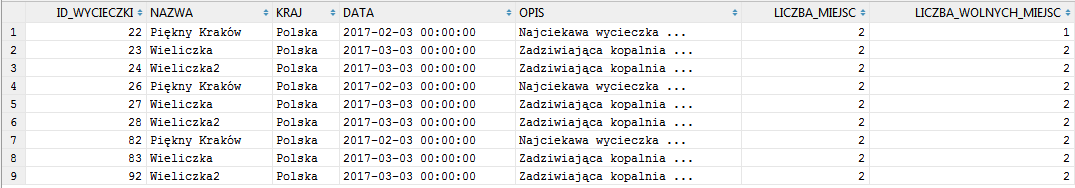
**CREATE or replace function** *dostepne\_wycieczki*(fkraj **in VARCHAR2**,

fdata\_od **in DATE**, fdata\_do **in DATE**)  
 **return sys\_refcursor  
 is** datecheck **NUMBER**;  
 **out sys\_refcursor**;  
 **begin  
 if** fdata\_do - fdata\_od < 0 **then  
 raise** value\_error ;  
 **end if**;  
 **open out for  
 Select** *\** **from** WYCIECZKI w

**where** w.**KRAJ** = fkraj **and** w.**DATA** - fdata\_do <= 0

**and** w.**DATA** - fdata\_od >=0;  
 **return out**;  
 **end**;

**declare  
 result sys\_refcursor**;  
 FKRAJ **VARCHAR2**(4000) := **'Polska'**;  
 FDATA\_OD **DATE** := *TO\_DATE*(**'10102015'**,**'DDMMYYYY'**);  
 FDATA\_DO **DATE** := *TO\_DATE*(**'10102018'**,**'DDMMYYYY'**);  
**begin  
 result** := *DOSTEPNE\_WYCIECZKI*(  
 FKRAJ => FKRAJ,  
 FDATA\_OD => FDATA\_OD,  
 FDATA\_DO => FDATA\_DO  
 );  
 ? := **result**;  
**end**;



5. Tworzenie procedur modyfikujących dane. Należy przygotować zestaw procedur pozwalających na modyfikację danych oraz kontrolę poprawności ich wprowadzania

a) dodaj\_rezerwacje(id\_wycieczki, id\_osoby), procedura powinna kontrolować czy wycieczka jeszcze się nie odbyła, i czy sa wolne miejsca

**CREATE or REPLACE procedure** *dodaj\_rezerwacje*(id\_wycieczki1 **NUMBER**, id\_osoby **NUMBER**)  
 **is** condition\_trip\_check **NUMBER**;  
 condition\_person\_check **NUMBER**;  
 condition\_free\_places **NUMBER**;  
 **begin  
 SELECT** *count*(*\**) **into** condition\_person\_check **from** OSOBY **where** OSOBY.**ID\_OSOBY** = **id\_osoby**;  
 **SELECT** *count*(*\**) **into** condition\_trip\_check **from** WYCIECZKI w **where** w.**ID\_WYCIECZKI** = id\_wycieczki1 **and** w.**DATA** - *sysdate* > 0;  
 **SELECT** *count*(*\**) **into** condition\_free\_places **from** wycieczki\_miejsca wm **where** wm.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **if** condition\_person\_check > 0 **and** condition\_trip\_check > 0 **and** condition\_free\_places > 0 **then  
 INSERT INTO** REZERWACJE (**id\_wycieczki**, **id\_osoby**, **status**)  
 **VALUES**(id\_wycieczki1,**id\_osoby**,**'N'**);  
 **commit**;  
 **end if**;  
 **end**;

b) zmien\_status\_rezerwacji(id\_rezerwacji, status), procedura kontrolować czy możliwa jest zmiana statusu, np. zmiana statusu już anulowanej wycieczki (przywrócenie do stanu aktywnego nie zawsze jest możliwe)

**CREATE or REPLACE procedure** *zmien\_status\_rezerwacji*(id\_rezerwacji1 **NUMBER**, status1 **CHARACTER**)  
 **is***--sprawdzamy czy wcieczka jeszcze sie nie odbyla i czy ma wolen miejsca* id\_wycieczki1 **number**;  
 condition\_reservation\_check **NUMBER**;  
 condition\_free\_places **NUMBER**;  
 condition\_is\_valid **NUMBER**;  
 **begin  
 SELECT** r.**ID\_WYCIECZKI into** id\_wycieczki1 **from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawadzamy czy sa jeszcze wolne miejsca na tej wycieczce* **SELECT "liczba wolnych miejsc" into** condition\_free\_places **from** wycieczki\_miejsca wm **where** wm.**ID\_WYCIECZKI** = id\_wycieczki1; *--ok  
 --spawdzamy czy istnieje taka rezerwacja* **SELECT** *count*(*\**) **into** condition\_reservation\_check **from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawdamy czy ta wycieczka jeszcze sie nie odbyla* **SELECT** *count*(*\**) **into** condition\_is\_valid **from** WYCIECZKI w  
 **where** w.**DATA** - *sysdate* > 0 **and** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 *--sprawdzamy wartosci naszych wywolan oraz wprowadzony status  
 --zmiana statusy na nowa mozliwa gdy sa miejsca  
 --na A,P mozemy zmienic zawsze jezeli wycieczka jeszcze sie nie odbyla* **if** ((status1 = **'N'**) **and** condition\_free\_places>0 **and** condition\_is\_valid> 0) **or** ((status1 = **'A' or** status1 = **'P' or** status1 = **'Z'**) **and** condition\_is\_valid > 0) **then  
 BEGIN**

**UPDATE** REZERWACJE r **SET** r.**STATUS** = status1

**where** r.**NR\_REZERWACJI** = id\_rezerwacji1;  
 **commit** ;  
 **end**;  
 **ELSE  
 raise** VALUE\_ERROR ;  
 **end if**;  
 **end**;

c) zmien\_liczbe\_miejsc(id\_wycieczki, liczba\_miejsc), nie wszystkie zmiany liczby miejsc są dozwolone, nie można zmniejszyć liczby miesc na wartość poniżej liczby zarezerwowanych miejsc

**CREATE or REPLACE PROCEDURE** *zmien\_liczbe\_miejsc*(id\_wycieczki1 **NUMBER**,

liczba\_miejsc1 **NUMBER**)  
**is** is\_trip\_valid **NUMBER**;  
 participants **NUMBER**;  
**begin** *--sprawdzamy czy wycieczka jeszcze sie nie odbyla i czy liczba zarezerwowanych miejsc nie jest wieksza od tej na ktora chcemy zmienic* **SELECT** *count*(*\**) **into** participants **FROM** WYCIECZKI w  
 **JOIN** REZERWACJE r **ON** w.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI**

**and** r.**STATUS** != **'A'  
 JOIN** OSOBY o **ON** r.**ID\_OSOBY** = o.**ID\_OSOBY  
 where** id\_wycieczki1 = w.**ID\_WYCIECZKI**;  
 **SELECT** *count*(*\**) **into** is\_trip\_valid **from** WYCIECZKI w

**where** w.**ID\_WYCIECZKI** = id\_wycieczki1 **and** w.**DATA** - *sysdate* > 0;  
 **if**(participants <= liczba\_miejsc1 **and** is\_trip\_valid > 0) **then  
 UPDATE** WYCIECZKI w **SET** w.**LICZBA\_MIEJSC** = liczba\_miejsc1

**where** w.**ID\_WYCIECZKI** = id\_wycieczki1; **COMMIT**;  
 **ELSE  
 raise** VALUE\_ERROR ;  
 **end if**;  
**end**;

6. Dodajemy tabelę dziennikującą zmiany statusu rezerwacji

rezerwacje\_log(id, id\_rezerwacji, data, status) Należy zmienić warstwę procedur modyfikujących dane tak aby dopisywały informację do dziennika

--tworzenie tabeli dziennikującej

**CREATE TABLE** REZERWACJE\_LOG  
(  
 **ID INT GENERATED ALWAYS AS IDENTITY NOT NULL**,  
 **ID\_REZERWACJI INT**,  
 **DATA DATE default** *sysdate* **not null**,  
 **STATUS CHAR**(1), *--poprzednia wartosc* **CONSTRAINT** REZERWACJE\_LOG\_PK **PRIMARY KEY** (  
 **ID** )  
 **ENABLE**);

--dodanie do tabeli rezerwacje\_log relacji z tabela rezerwacje

**ALTER TABLE** REZERWACJE\_LOG  
**ADD CONSTRAINT** REZERWACJE\_LOG\_FK1 **FOREIGN KEY** (  
 **ID\_REZERWACJI** )  
 **REFERENCES** REZERWACJE  
 (  
 **NR\_REZERWACJI** )  
 **ENABLE**;

--dodanie check’a na statusie w tabeli rezerwacje\_log

**ALTER TABLE** REZERWACJE\_LOG  
**ADD CONSTRAINT** REZERWACJE\_LOG\_CHK1 **CHECK**(**STATUS in** (**'N'**,**'P'**,**'Z'**,**'A'**))  
 **ENABLE** ;  
  
**commit** ;

--modyfikacja procedury

**CREATE or REPLACE procedure** *zmien\_status\_rezerwacji*(id\_rezerwacji1 **NUMBER**, status1 **CHARACTER**)  
 **is***--sprawdzamy czy wcieczka jeszcze sie nie odbyla i czy ma wolen miejsca* id\_wycieczki1 **number**;  
 condition\_reservation\_check **NUMBER**;  
 condition\_free\_places **NUMBER**;  
 condition\_is\_valid **NUMBER**;  
 prev\_status **CHARACTER**;  
 **begin  
 SELECT** r.**ID\_WYCIECZKI into** id\_wycieczki1 **from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawadzamy czy sa jeszcze wolne miejsca na tej wycieczce* **SELECT "liczba wolnych miejsc" into** condition\_free\_places **from** wycieczki\_miejsca wm **where** wm.**ID\_WYCIECZKI** = id\_wycieczki1; *--ok  
 --spawdzamy czy istnieje taka rezerwacja* **SELECT** *count*(*\**) **into** condition\_reservation\_check **from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawdamy czy ta wycieczka jeszcze sie nie odbyla* **SELECT** *count*(*\**) **into** condition\_is\_valid **from** WYCIECZKI w  
 **where** w.**DATA** - *sysdate* > 0 **and** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 *--sprawdzamy wartosci naszych wywolan oraz wprowadzony status  
 --zmiana statusy na nowa mozliwa gdy sa miejsca  
 --na A,P mozemy zmienic zawsze jezeli wycieczka jeszcze sie nie odbyla* **if** ((status1 = **'N'**) **and** condition\_free\_places>0 **and** condition\_is\_valid> 0) **or** ((status1 = **'A' or** status1 = **'P' or** status1 = **'Z'**) **and** condition\_is\_valid > 0) **then  
 BEGIN  
 SELECT status into** prev\_status **from** REZERWACJE **where** id\_rezerwacji1 = REZERWACJE.**NR\_REZERWACJI**;  
 **UPDATE** REZERWACJE r **SET** r.**STATUS** = status1 **where** r.**NR\_REZERWACJI** = id\_rezerwacji1;  
 **INSERT INTO** REZERWACJE\_LOG (**ID\_REZERWACJI**, **STATUS**) **VALUES** (id\_rezerwacji1,prev\_status);  
 **commit** ;  
 **end**;  
 **ELSE  
 raise** VALUE\_ERROR ;  
 **end if**;  
 **end**;

7. Zmiana struktury bazy danych, w tabeli wycieczki dodajemy redundantne pole liczba\_wolnych\_miejsc

**ALTER TABLE** WYCIECZKI  
 **ADD** (**liczba\_wolnych\_miejsc NUMBER default null**);  
**commit**;

Należy zmodyfikować zestaw widoków. Proponuję dodać kolejne widoki (np. z sufiksem 2), które pobierają informację o wolnych miejscach z nowo dodanego pola.

**CREATE VIEW** wycieczki\_osoby2  
 **AS  
 SELECT** w.**ID\_WYCIECZKI**,  
 w.**NAZWA**,  
 w.**KRAJ**,  
 w.**DATA**,  
 o.**IMIE**,  
 o.**NAZWISKO**,  
 r.**STATUS**,  
 w.**liczba\_wolnych\_miejsc  
 FROM** WYCIECZKI w  
 **JOIN** REZERWACJE r **ON** w.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI  
 JOIN** OSOBY o **ON** r.**ID\_OSOBY** = o.**ID\_OSOBY**;

**CREATE VIEW** wycieczki\_przyszle2  
 **AS  
 SELECT** w.**KRAJ**,  
 w.**DATA**,  
 w.**NAZWA**,  
 o.**IMIE**,  
 o.**NAZWISKO**,  
 r.**STATUS**,  
 w.**liczba\_wolnych\_miejsc  
 FROM** wycieczki w  
 **JOIN** REZERWACJE r **on** w.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI and** r.**STATUS !**= **'A'  
 JOIN** OSOBY o **on** r.**ID\_OSOBY** = o.**ID\_OSOBY  
 where** w.**DATA** - *SYSDATE* > 0;

**CREATE or replace view** wycieczki\_miejsca2  
 **as  
 SELECT** w.**ID\_WYCIECZKI**,  
 w.**KRAJ**,  
 w.**DATA**,  
 w.**NAZWA**,  
 w.**LICZBA\_MIEJSC**,  
 w.**liczba\_wolnych\_miejsc  
 from** wycieczki w;  
 **commit** ;  
**SELECT** *\** **from** wycieczki\_miejsca;

**CREATE view** dostępne\_wycieczki2  
 **as  
 SELECT** w.**KRAJ**,  
 w.**DATA**,  
 w.**NAZWA**,  
 w.**LICZBA\_MIEJSC**,  
 w.**liczba\_wolnych\_miejsc  
 from** wycieczki w  
 **where  
 liczba\_wolnych\_miejsc** > 0 **and** w.**DATA** - *SYSDATE* > 0;  
  
**commit**;

Należy napisać procedurę przelicz która zaktualizuje wartość liczby wolnych miejsc dla już istniejących danych Należy zmodyfikować warstwę procedur pobierających dane, podobnie jak w przypadku widoków.

**CREATE or REPLACE procedure** *AKTUALIZACJA\_WOLNYCH\_MIEJSC* **as  
 begin  
 UPDATE** WYCIECZKI w  
 **SET** w.**liczba\_wolnych\_miejsc** =  
 w.**LICZBA\_MIEJSC** - (  
 **select** *count*(*\**) **FROM** wycieczki w1  
 **JOIN** REZERWACJE r **on** w1.**ID\_WYCIECZKI** = r.**ID\_WYCIECZKI  
 JOIN** OSOBY o **on** r.**ID\_OSOBY** = o.**ID\_OSOBY  
 where** r.**STATUS** != **'A' and** w1.**ID\_WYCIECZKI** = w.**ID\_WYCIECZKI** )  
 **where** w.**liczba\_wolnych\_miejsc is null**;  
 **end**;  
 **commit** ;

Należy zmodyfikować procedury wprowadzające dane tak aby korzystały/aktualizowały pole liczba \_wolnych\_miejsc w tabeli wycieczki Najlepiej to zrobić tworząc nowe wersje (np. z sufiksem 2)

A) **CREATE or REPLACE procedure** *dodaj\_rezerwacje2*(id\_wycieczki1 **NUMBER**, id\_osoby **NUMBER**)  
 **is** condition\_trip\_check **NUMBER**;  
 condition\_person\_check **NUMBER**;  
 condition\_free\_places **NUMBER**;  
 **begin  
 SELECT** *count*(*\**) **into** condition\_person\_check **from** OSOBY **where** OSOBY.**ID\_OSOBY** = **id\_osoby**;  
 **SELECT** *count*(*\**) **into** condition\_trip\_check **from** WYCIECZKI w **where** w.**ID\_WYCIECZKI** = id\_wycieczki1 **and** w.**DATA** - *sysdate* > 0;  
 **SELECT** wm.**liczba\_wolnych\_miejsc into** condition\_free\_places **from** WYCIECZKI wm **where** wm.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **if** condition\_person\_check > 0 **and** condition\_trip\_check > 0 **and** condition\_free\_places > 0 **then  
 INSERT INTO** REZERWACJE (**id\_wycieczki**, **id\_osoby**, **status**)  
 **VALUES**(id\_wycieczki1,**id\_osoby**,**'N'**);  
 **UPDATE** WYCIECZKI w **SET** w.**liczba\_wolnych\_miejsc** = w.**liczba\_wolnych\_miejsc** - 1 **where** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **commit**;  
 **end if**;  
 **end**;  
**commit** ;

B)

**CREATE or REPLACE procedure** *zmien\_status\_rezerwacji2*(id\_rezerwacji1 **NUMBER**, status1 **CHARACTER**)  
 **is***--sprawdzamy czy wcieczka jeszcze sie nie odbyla i czy ma wolen miejsca* id\_wycieczki1 **number**;  
 condition\_reservation\_check **NUMBER**;  
 condition\_free\_places **NUMBER**;  
 condition\_is\_valid **NUMBER**;  
 prev\_status **CHARACTER**;  
 **begin  
 SELECT** r.**ID\_WYCIECZKI into** id\_wycieczki1 **from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawadzamy czy sa jeszcze wolne miejsca na tej wycieczce* **SELECT** wm.**liczba\_wolnych\_miejsc into** condition\_free\_places **from** WYCIECZKI wm **where** wm.**ID\_WYCIECZKI** = id\_wycieczki1; *--ok  
 --spawdzamy czy istnieje taka rezerwacja* **SELECT** *count*(*\**) **into** condition\_reservation\_check **from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawdamy czy ta wycieczka jeszcze sie nie odbyla* **SELECT** *count*(*\**) **into** condition\_is\_valid **from** WYCIECZKI w **where** w.**DATA** - *sysdate* > 0 **and** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 *--sprawdzamy wartosci naszych wywolan oraz wprowadzony status  
 --zmiana statusy na nowa mozliwa gdy sa miejsca  
 --na A,P mozemy zmienic zawsze jezeli wycieczka jeszcze sie nie odbyla* **if** ((status1 = **'N'**) **and** condition\_free\_places>0 **and** condition\_is\_valid> 0) **or** ((status1 = **'A' or** status1 = **'P' or** status1 = **'Z'**) **and** condition\_is\_valid > 0) **then  
 BEGIN  
 SELECT status into** prev\_status **from** REZERWACJE **where** id\_rezerwacji1 = REZERWACJE.**NR\_REZERWACJI**;  
 **UPDATE** REZERWACJE r **SET** r.**STATUS** = status1 **where** r.**NR\_REZERWACJI** = id\_rezerwacji1;  
 **INSERT INTO** REZERWACJE\_LOG (**ID\_REZERWACJI**, **STATUS**) **VALUES** (id\_rezerwacji1,prev\_status);  
 **if** prev\_status != **'A' and** status1 = **'A' then  
 UPDATE** WYCIECZKI w **SET** w.**liczba\_wolnych\_miejsc** = w.**liczba\_wolnych\_miejsc** + 1 **where** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **end if**;  
 **commit** ;  
 **end**;  
 **ELSE  
 raise** VALUE\_ERROR ;  
 **end if**;  
 **end**;

**C)**

**CREATE or REPLACE PROCEDURE** *zmien\_liczbe\_miejsc2*(id\_wycieczki1 **NUMBER**, liczba\_miejsc1 **NUMBER**)  
**is** is\_trip\_valid **NUMBER**;  
 participants **NUMBER**;  
**begin** *--sprawdzamy czy wycieczka jeszcze sie nie odbyla i czy liczba zarezerwowanych miejsc nie jest wieksza od tej na ktora chcemy zmienic* **SELECT** w.**LICZBA\_MIEJSC**-w.**liczba\_wolnych\_miejsc into** participants **FROM** WYCIECZKI w **where** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **SELECT** *count*(*\**) **into** is\_trip\_valid **from** WYCIECZKI w **where** w.**ID\_WYCIECZKI** = id\_wycieczki1 **and** w.**DATA** - *sysdate* > 0;  
 **if**(participants <= liczba\_miejsc1 **and** is\_trip\_valid > 0) **then  
 UPDATE** WYCIECZKI w **SET** w.**LICZBA\_MIEJSC** = liczba\_miejsc1 **where** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **UPDATE** WYCIECZKI w **SET** w.**LICZBA\_WOLNYCH\_MIEJSC** = w.**LICZBA\_MIEJSC** - participants **where** w.**ID\_WYCIECZKI** = id\_wycieczki1; **COMMIT**;  
 **ELSE  
 raise** VALUE\_ERROR ;  
 **end if**;  
**end**;

Komentarz: Dodanie jednego prostego pola do tabeli powoduje dużo zmian nawet w nieskomplikowanej bazie danych

8. Zmiana strategii zapisywania do dziennika rezerwacji. Realizacja przy pomocy triggerów Należy wprowadzić zmianę która spowoduje że zapis do dziennika rezerwacji będzie realizowany przy pomocy trierów

triger obsługujący dodanie rezerwacji

**CREATE or replace TRIGGER** triger\_obslugujacy\_dodanie\_rezerwacji  
 **AFTER INSERT on** REZERWACJE  
 **for each row  
 BEGIN  
 insert into** REZERWACJE\_LOG ( **ID\_REZERWACJI**, **DATA**, **STATUS**) **VALUES** (:**new**.**NR\_REZERWACJI**,*sysdate*,:**new**.**STATUS**);  
 **end**;  
**commit**;

triger obsługujący zmianę statusu

**CREATE or replace TRIGGER** triger\_obslugujacy\_zmiane\_statusu  
 **BEFORE UPDATE on** REZERWACJE  
 **for each row  
 BEGIN  
 insert into** REZERWACJE\_LOG ( **ID\_REZERWACJI**, **DATA**, **STATUS**) **VALUES** (:**new**.**NR\_REZERWACJI**,*sysdate*,:**old**.**STATUS**);  
 **end**;  
**commit**;

triger zabraniający usunięcia rezerwacji

**CREATE or replace TRIGGER** triger\_zabraniajacy\_usuniecia\_rezerwacji  
 **BEFORE DELETE on** REZERWACJE  
 **for each row  
 BEGIN** *raise\_application\_error*(-20001,**'Rezerwacja nie moze zostać usunięta!!'**);  
 **end**;  
**commit**;

Oczywiście po wprowadzeniu tej zmiany należy uaktualnić procedury modyfikujące dane. Najlepiej to zrobić tworząc nowe wersje (np. z sufiksem 3)

A)

**CREATE or REPLACE procedure** *dodaj\_rezerwacje3*(id\_wycieczki1 **NUMBER**, id\_osoby **NUMBER**)  
 **is** condition\_trip\_check **NUMBER**;  
 condition\_person\_check **NUMBER**;  
 condition\_free\_places **NUMBER**;  
 **begin  
 SELECT** *count*(*\**) **into** condition\_person\_check **from** OSOBY **where** OSOBY.**ID\_OSOBY** = **id\_osoby**;  
 **SELECT** *count*(*\**) **into** condition\_trip\_check **from** WYCIECZKI w **where** w.**ID\_WYCIECZKI** = id\_wycieczki1 **and** w.**DATA** - *sysdate* > 0;  
 **SELECT** wm.**liczba\_wolnych\_miejsc into** condition\_free\_places **from** WYCIECZKI wm **where** wm.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **if** condition\_person\_check > 0 **and** condition\_trip\_check > 0 **and** condition\_free\_places > 0 **then  
 INSERT INTO** REZERWACJE (**id\_wycieczki**, **id\_osoby**, **status**)  
 **VALUES**(id\_wycieczki1,**id\_osoby**,**'N'**);  
 **UPDATE** WYCIECZKI w **SET** w.**liczba\_wolnych\_miejsc** = w.**liczba\_wolnych\_miejsc** - 1 **where** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **commit**;  
 **end if**;  
 **end**;  
**commit** ;

B)

**CREATE or REPLACE procedure** *zmien\_status\_rezerwacji3*(id\_rezerwacji1 **NUMBER**, status1 **CHARACTER**)  
 **is***--sprawdzamy czy wcieczka jeszcze sie nie odbyla i czy ma wolen miejsca* id\_wycieczki1 **number**;  
 condition\_reservation\_check **NUMBER**;  
 condition\_free\_places **NUMBER**;  
 condition\_is\_valid **NUMBER**;  
 prev\_status **CHARACTER**;  
 **begin  
 SELECT** r.**ID\_WYCIECZKI into** id\_wycieczki1 **from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawadzamy czy sa jeszcze wolne miejsca na tej wycieczce* **SELECT** wm.**liczba\_wolnych\_miejsc into** condition\_free\_places **from** WYCIECZKI wm **where** wm.**ID\_WYCIECZKI** = id\_wycieczki1; *--ok  
 --spawdzamy czy istnieje taka rezerwacja* **SELECT** *count*(*\**) **into** condition\_reservation\_check **from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawdamy czy ta wycieczka jeszcze sie nie odbyla* **SELECT** *count*(*\**) **into** condition\_is\_valid **from** WYCIECZKI w **where** w.**DATA** - *sysdate* > 0 **and** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 *--sprawdzamy wartosci naszych wywolan oraz wprowadzony status  
 --zmiana statusy na nowa mozliwa gdy sa miejsca  
 --na A,P mozemy zmienic zawsze jezeli wycieczka jeszcze sie nie odbyla* **if** ((status1 = **'N'**) **and** condition\_free\_places>0 **and** condition\_is\_valid> 0) **or** ((status1 = **'A' or** status1 = **'P' or** status1 = **'Z'**) **and** condition\_is\_valid > 0) **then  
 BEGIN  
 SELECT status into** prev\_status **from** REZERWACJE **where** id\_rezerwacji1 = REZERWACJE.**NR\_REZERWACJI**;  
 **UPDATE** REZERWACJE r **SET** r.**STATUS** = status1 **where** r.**NR\_REZERWACJI** = id\_rezerwacji1;  
 **if** prev\_status != **'A' and** status1 = **'A' then  
 UPDATE** WYCIECZKI w **SET** w.**liczba\_wolnych\_miejsc** = w.**liczba\_wolnych\_miejsc** + 1 **where** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **end if**;  
 **commit** ;  
 **end**;  
 **ELSE  
 raise** VALUE\_ERROR ;  
 **end if**;  
 **end**;

Komentarz: Zastosowanie triggerow do osługi tabeli logów pozwoliło na zautomatyzowanie jej obsługi.

9. Zmiana strategii obsługi redundantnego pola liczba\_wolnych\_miejsc. Realizacja przy pomocy trigerów

Triger obsługujący dodanie rezerwacji

**CREATE or replace TRIGGER** triger\_obslugujacy\_dodanie\_rezerwacji\_RWM  
 **AFTER INSERT on** REZERWACJE  
 **for each row  
 declare** tliczba\_wolnych\_miejc **NUMBER**;  
 **BEGIN  
 SELECT liczba\_wolnych\_miejsc into** tliczba\_wolnych\_miejc

**from** WYCIECZKI **where ID\_WYCIECZKI** = :**NEW**.**ID\_WYCIECZKI**;  
 **if** tliczba\_wolnych\_miejc < 1 **then** *raise\_application\_error*(-20001,**'Nie ma wolnych miesjc'**);  
 **end if**;  
 **UPDATE** WYCIECZKI

**SET liczba\_wolnych\_miejsc** = **liczba\_wolnych\_miejsc** - 1

**WHERE ID\_WYCIECZKI** = :**NEW**.**ID\_WYCIECZKI**;  
 **end**;  
 **commit**;

Triger obsługujący zmianę statusu

**CREATE or replace TRIGGER** triger\_obslugujacy\_zmiane\_statusu\_RWM  
 **AFTER UPDATE on** REZERWACJE  
 **for each row  
 declare** tliczba\_wolnych\_miejc **NUMBER**;  
 **BEGIN  
 SELECT liczba\_wolnych\_miejsc into** tliczba\_wolnych\_miejc

**from** WYCIECZKI **where ID\_WYCIECZKI** = :**NEW**.**ID\_WYCIECZKI**;  
 **if** (tliczba\_wolnych\_miejc < 1 **and** :**OLD**.**STATUS** = **'A'**

**AND** :**NEW**.**STATUS** != **'A'**) **then** *raise\_application\_error*(-20001,**'Nie ma wolnych miesjc'**);  
 **end if**;  
 **IF** :**OLD**.**STATUS** = **'A' AND** :**NEW**.**STATUS** != **'A' THEN  
 UPDATE** WYCIECZKI W

**SET** W.**liczba\_wolnych\_miejsc** = W.**liczba\_wolnych\_miejsc** - 1

**WHERE** W.**ID\_WYCIECZKI** = :**NEW**.**ID\_WYCIECZKI**;  
 **END IF**;  
 **IF** :**OLD**.**STATUS** != **'A' AND** :**NEW**.**STATUS** = **'A' THEN  
 UPDATE** WYCIECZKI W

**SET** W.**liczba\_wolnych\_miejsc** = W.**liczba\_wolnych\_miejsc** + 1

**WHERE** W.**ID\_WYCIECZKI** = :**NEW**.**ID\_WYCIECZKI**;  
 **END IF**;  
 **end**;

Triger obsługujący zmianę liczby miejsc na poziomie wycieczki

**CREATE or replace TRIGGER** triger\_obslugujacy\_zmiane\_liczby\_miejsc\_wycieczki\_RWM  
 **before UPDATE on** WYCIECZKI  
 **for each row  
 BEGIN  
 if** (:**NEW**.**LICZBA\_MIEJSC** - (:**OLD**.**LICZBA\_MIEJSC** - :**OLD**.**liczba\_wolnych\_miejsc**)) < 0 **then** *raise\_application\_error*(-20001,**'Zla zmiana liczy miejsc'**);  
 **end if**;  
 :**NEW**.**liczba\_wolnych\_miejsc** := :**NEW**.**LICZBA\_MIEJSC** - (:**OLD**.**LICZBA\_MIEJSC** - :**OLD**.**liczba\_wolnych\_miejsc**);  
 *--UPDATE WYCIECZKI W SET W.liczba\_wolnych\_miejsc = :NEW.LICZBA\_MIEJSC - (:OLD.LICZBA\_MIEJSC - :OLD.liczba\_wolnych\_miejsc) WHERE W.ID\_WYCIECZKI = :NEW.ID\_WYCIECZKI;* **end**;

Oczywiście po wprowadzeniu tej zmiany należy uaktualnić procedury modyfikujące dane. Najlepiej to zrobić tworząc nowe wersje (np. z sufiksem 3)

**CREATE or REPLACE procedure** *dodaj\_rezerwacje4*(id\_wycieczki1 **NUMBER**, id\_osoby **NUMBER**)  
 **is** condition\_trip\_check **NUMBER**;  
 condition\_person\_check **NUMBER**;  
 **begin  
 SELECT** *count*(*\**) **into** condition\_person\_check

**from** OSOBY **where** OSOBY.**ID\_OSOBY** = **id\_osoby**;  
 **SELECT** *count*(*\**) **into** condition\_trip\_check **from** WYCIECZKI w

**where** w.**ID\_WYCIECZKI** = id\_wycieczki1 **and** w.**DATA** - *sysdate* > 0;  
 **if** condition\_person\_check > 0 **and** condition\_trip\_check > 0 **then  
 INSERT INTO** REZERWACJE (**id\_wycieczki**, **id\_osoby**, **status**)  
 **VALUES**(id\_wycieczki1,**id\_osoby**,**'N'**);  
 **commit**;  
 **end if**;  
 **end**;

b)

**CREATE or REPLACE procedure** *zmien\_status\_rezerwacji4*(id\_rezerwacji1 **NUMBER**, status1 **CHARACTER**)  
 **is***--sprawdzamy czy wcieczka jeszcze sie nie odbyla i czy ma wolen miejsca* id\_wycieczki1 **number**;  
 condition\_reservation\_check **NUMBER**;  
 condition\_is\_valid **NUMBER**;  
 prev\_status **CHARACTER**;  
 **begin  
 SELECT** r.**ID\_WYCIECZKI into** id\_wycieczki1

**from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok* **SELECT** *count*(*\**) **into** condition\_reservation\_check

**from** REZERWACJE r **where** r.**NR\_REZERWACJI** = id\_rezerwacji1; *--ok  
 --sprawdamy czy ta wycieczka jeszcze sie nie odbyla* **SELECT** *count*(*\**) **into** condition\_is\_valid **from** WYCIECZKI w

**where** w.**DATA** - *sysdate* > 0 **and** w.**ID\_WYCIECZKI** = id\_wycieczki1;  
 **if** condition\_is\_valid> 0 **and** condition\_reservation\_check > 0 **then  
 BEGIN  
 UPDATE** REZERWACJE r **SET** r.**STATUS** = status1

**where** r.**NR\_REZERWACJI** = id\_rezerwacji1;  
 **commit** ;  
 **end**;  
 **ELSE  
 raise** VALUE\_ERROR ;  
 **end if**;  
 **end**;

**CREATE or REPLACE PROCEDURE** *zmien\_liczbe\_miejsc4*(id\_wycieczki1 **NUMBER**, liczba\_miejsc1 **NUMBER**)  
**is** is\_trip\_valid **NUMBER**;  
**begin** *--sprawdzamy czy wycieczka jeszcze sie nie odbyla i czy liczba zarezerwowanych miejsc nie jest wieksza od tej na ktora chcemy zmienic* **SELECT** *count*(*\**) **into** is\_trip\_valid **from** WYCIECZKI w

**where** w.**ID\_WYCIECZKI** = id\_wycieczki1 **and** w.**DATA** - *sysdate* > 0;  
 **if**(is\_trip\_valid > 0) **then  
 UPDATE** WYCIECZKI w **SET** w.**LICZBA\_MIEJSC** = liczba\_miejsc1

**where** w.**ID\_WYCIECZKI** = id\_wycieczki1; **COMMIT**;  
 **ELSE  
 raise** VALUE\_ERROR ;  
 **end if**;  
**end**;

Komentarz: Zastosowanie triggerow pozwoliło na tworzenie czytelniejszych oraz krótszych procedur.