

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', 'Leczyński', '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_miejscal 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 UCZESTNICZY_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 UCZESTNICZY_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_rezerwacje_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_rezerwacje_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;
```