

# Gracjan Filipek – raport, Oracle PL/SQL

## Programowanie proceduralne - widoki, procedury, triggerzy itp.

### I. Oracle PL/SQL

#### 1. Tabele

Trip (trip\_id, name, country, date, no\_places)

Person(person\_id, firstname, lastname)

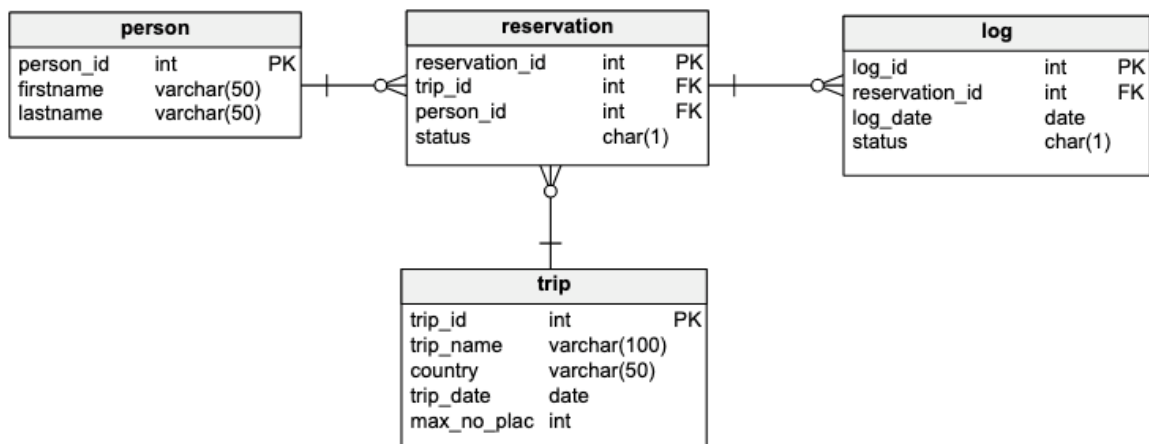
Reservation(reservation\_id, trip\_id, person\_id, status)

Pole status w tabeli Rezerwacje może przyjmować jedną z 4 wartości

N – New

P – Paid

C – Canceled



#### 1a) Wypełnianie tabeli przykładowymi danymi

Zmodyfikuj początkowy schemat bazy danych dodając tabelę "słownikową" z listą krajów

### Kod

```
create table country
(
    country_name varchar(50),
```

```
constraint coutry_pk primary key (country_name) enable
);
```

## 2. Wypełnianie tabeli przykładowymi danymi

4 wycieczki

10 osób

10 rezerwacji

### ### Kod

```
-- country
insert into COUNTRY(COUNTRY_NAME)
values ('Polska');
insert into COUNTRY(COUNTRY_NAME)
values ('Francja');
insert into COUNTRY(COUNTRY_NAME)
values ('Hiszpania');
insert into COUNTRY(COUNTRY_NAME)
values ('Grecja');

-- trip
insert into trip(trip_name, country, trip_date, max_no_places)
values ('Wycieczka do Paryża', 'Francja', to_date('2022-09-12','YYYY-MM-DD'), 3);

insert into trip(trip_name, country, trip_date, max_no_places)
values ('Piękny Kraków', 'Polska', to_date('2023-07-03','YYYY-MM-DD'), 2);

insert into trip(trip_name, country, trip_date, max_no_places)
values ('Znów do Francji', 'Francja', to_date('2023-05-01','YYYY-MM-DD'), 5);

insert into trip(trip_name, country, trip_date, max_no_places)
values ('Hel', 'Polska', to_date('2023-05-01','YYYY-MM-DD'), 7);

-- person
insert into person(firstname, lastname)
values ('Jan', 'Nowak');

insert into person(firstname, lastname)
values ('Jan', 'Kowalski');

insert into person(firstname, lastname)
values ('Jan', 'Nowakowski');

insert into person(firstname, lastname)
values ('Adam', 'Kowalski');

insert into person(firstname, lastname)
values ('Novak', 'Nowak');

insert into person(firstname, lastname)
values ('Piotr', 'Piotrowski');

insert into PERSON(firstname, lastname)
VALUES ('Marek', 'Adamowski');

insert into PERSON(firstname, lastname)
```

```

VALUES ('Adam', 'Markowski');

insert into PERSON(firstname, lastname)
VALUES ('Filip', 'Maciejewski');

insert into PERSON(firstname, lastname)
VALUES ('Maciej', 'Filipowski');

-- reservation
-- trip 1
insert into reservation(trip_id, person_id, status)
values (1, 1, 'P');

insert into reservation(trip_id, person_id, status)
values (1, 2, 'N');

-- trip 2
insert into reservation(trip_id, person_id, status)
values (2, 1, 'C');

insert into reservation(trip_id, person_id, status)
values (2, 4, 'C');

insert into reservation(trip_id, person_id, status)
values (2, 7, 'N');

insert into reservation(trip_id, person_id, status)
values (2, 9, 'P');

-- trip 3
insert into reservation(trip_id, person_id, status)
values (3, 4, 'P');

-- trip 4
insert into reservation(trip_id, person_id, status)
values (4, 6, 'P');

insert into reservation(trip_id, person_id, status)
values (4, 9, 'C');

insert into reservation(trip_id, person_id, status)
values (4, 3, 'N');

```

3. Tworzenie widoków. Należy przygotować kilka widoków ułatwiających dostęp do danych. Należy zwrócić uwagę na strukturę kodu (należy unikać powielania kodu)

- a) Reservations(country,trip\_date,trip\_name, firstname, lastname,reservation\_id,status)
- b) Trips(country,trip\_date, trip\_name,no\_places, no\_available\_places)
- c) AvailableTrips(country,trip\_date, trip\_name,no\_places, no\_available\_places)

Proponowany zestaw widoków można rozbudować wedle uznania/potrzeb

- np. można dodać nowe/pomocnicze widoki
- np. można zmienić def. widoków, dodając nowe/potrzebne pola

### Komentarz

Widoki zostały przeze mnie lekko zmodyfikowane (Reservations zawiera dodatkowo trip\_id oraz person\_id), gdyż w takiej formie można je skutecznie zastosować w procedurach z zadania 4. Dodatkowo stworzyłem też widok pomocniczy, który zlicza liczbę rezerwacji dla każdej wycieczki.

### ### Kod

```
CREATE OR REPLACE VIEW Reservations (trip_id, country, trip_date,
trip_name, person_id, firstname, lastname, reservation_id, status) AS
select T.TRIP_ID, T.COUNTRY, T.TRIP_DATE, T.TRIP_NAME, P.PERSON_ID,
P.FIRSTNAME, P.LASTNAME, R.RESERVATION_ID, R.STATUS
from RESERVATION R
    join TRIP T on R.TRIP_ID = T.TRIP_ID
    join PERSON P on P.PERSON_ID = R.PERSON_ID;
```

	TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	PERSON_ID	FIRSTNAME	LASTNAME	RESERVATION_ID	STATUS
1	1	Francja	2022-09-12	Wycieczka do Paryza	1	Jan	Nowak	1	C
2	2	Polska	2023-07-03	Piękny Kraków	1	Jan	Nowak	3	C
3	1	Francja	2022-09-12	Wycieczka do Paryza	2	Jan	Kowalski	2	N
4	4	Polska	2023-05-01	Heł	3	Jan	Nowakowski	10	N
5	2	Polska	2023-07-03	Piękny Kraków	4	Adam	Kowalski	4	C
6	3	Francja	2023-05-01	Znów do Francji	4	Adam	Kowalski	7	P
7	4	Polska	2023-05-01	Heł	4	Adam	Kowalski	81	N
8	2	Polska	2023-07-03	Piękny Kraków	6	Piotr	Piotrowski	22	N
9	4	Polska	2023-05-01	Heł	6	Piotr	Piotrowski	8	P
10	2	Polska	2023-07-03	Piękny Kraków	7	Marek	Adamowski	5	N
11	3	Francja	2023-05-01	Znów do Francji	7	Marek	Adamowski	42	C
12	3	Francja	2023-05-01	Znów do Francji	8	Adam	Markowski	41	N
13	2	Polska	2023-07-03	Piękny Kraków	9	Filip	Maciejewski	6	P
14	4	Polska	2023-05-01	Heł	9	Filip	Maciejewski	9	C

```
/* auxiliary view */
CREATE OR REPLACE VIEW ActiveReservationsByTrips AS
select R.TRIP_ID, count(R.RESERVATION_ID) NUM_OF_RESERVATIONS
from RESERVATION R
where R.STATUS != 'C'
group by R.TRIP_ID;
```

	TRIP_ID	NUM_OF_RESERVATIONS
1	1	1
2	2	3
3	4	3
4	3	2

```
CREATE OR REPLACE VIEW Trips (trip_id, country, trip_date, trip_name,
no_places, num_of_available_places) AS
select T.TRIP_ID, T.COUNTRY, T.TRIP_DATE, T.TRIP_NAME, T.MAX_NO_PLACES,
(T.MAX_NO_PLACES - NVL(MR.NUM_OF_RESERVATIONS, 0))
from TRIP T
    left join ActiveReservationsByTrips MR on T.TRIP_ID = MR.TRIP_ID;
```

	TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES
1	1	Francja	2022-09-12	Wycieczka do Paryza	3	2
2	2	Polska	2023-07-03	Piękny Kraków	5	2
3	3	Francja	2023-05-01	Znów do Francji	2	0
4	4	Polska	2023-05-01	Heł	7	4
5	5	Hiszpania	2023-02-02	Czemu by nie?	5	5

```
CREATE OR REPLACE VIEW AvailableTrips (trip_id, country, trip_date,
trip_name, no_places, num_of_available_places) AS
select T.trip_id, T.COUNTRY, T.TRIP_DATE, T.TRIP_NAME, T.NO_PLACES,
T.NUM_OF_AVAILABLE_PLACES
from TRIPS T
```

```
where num_of_available_places > 0
and current_date < T.TRIP_DATE;
```

	TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES
1	2	Polska	2023-07-03	Piękny Kraków	5	2
2	4	Polska	2023-05-01	Hel	7	4

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

- TripParticipants(trip\_id), procedura ma zwracać podobny zestaw danych jak widok Reservations
- PersonReservations(person\_id), procedura ma zwracać podobny zestaw danych jak widok Reservations
- AvailableTrips(country, date\_from, date\_to)

Procedury/funkcje powinny zwracać tabelę/zbiór wyników

Należy zwrócić uwagę na kontrolę parametrów (np. jeśli parametrem jest trip\_id to należy sprawdzić czy taka wycieczka istnieje). Podobnie jak w przypadku widoków należy unikać powielania kodu.

### ### Komentarz

Pomijam wyniki działań funkcji pomocniczych, sprawdzających istnienie różnych elementów w tablicach, gdyż niewiele wniesie to do raportu oraz poprawność działania samych funkcji zwracających dane jest świadectwem poprawnego działania funkcji pomocniczych.

Zmieniłem też nazwę ostatniej funkcji na f\_available\_trips z powodu kolizji z widokiem o takiej samej nazwie

### ### Kod

```
create or replace type trip_participant as OBJECT
(
    reservation_id int,
    firstname      varchar2(50),
    lastname       varchar2(50),
    status         char
);

create or replace type trip_participant_table is table of trip_participant;

CREATE OR REPLACE FUNCTION trip_exists(trip_id in TRIP.trip_id%type)
RETURN boolean
AS
    exist number;
BEGIN
    select case
        when exists(select * from TRIP T where T.TRIP_ID =
trip_exists.trip_id) then 1
        else 0
        end
    into exist
    from dual;
    if exist = 1 then
```

```

        return true;
    else
        return false;
    end if;
END;

CREATE OR REPLACE FUNCTION trip_participants(trip_id TRIP.TRIP_ID%type)
RETURN trip_participant_table
AS
    result trip_participant_table;
BEGIN
    if not trip_exists(trip_id) then
        raise_application_error(-20001, 'trip with id ' || trip_id || ' not
found');
    end if;

    select trip_participant(RESERVATION_ID, FIRSTNAME, LASTNAME, STATUS)
bulk collect
    into result
    from RESERVATIONS R
    where R.TRIP_ID = trip_participants.trip_id
    and R.STATUS != 'C';
    return result;
END;

```

```

✓ select *
from trip_participants( trip_id: 4);
/* ##### */

```

trip\_participants

Output trip_participants				
	RESERVATION_ID	FIRSTNAME	LASTNAME	STATUS
1	8	Piotr	Piotrowski	P
2	10	Jan	Nowakowski	N
3	81	Adam	Kowalski	N

Gdy wycieczka nie istnieje:

```

48
49 ! select *
50 from trip_participants( trip_id: 20);

```

trip\_participants

```

[72000][20001]
ORA-20001: trip with id 20 not found
ORA-06512: przy "BD_409890.TRIP_PARTICIPANTS", linia 7
Position: 15

```

```

/* ##### */

```

```

create or replace type person_reservation as OBJECT
(
    reservation_id int,
    trip_name      varchar2(100),
    country        varchar2(50),

```

```

        trip_date      date,
        status         char
    );

create or replace type person_reservation_table is table of
person_reservation;

drop function person_exists;
CREATE OR REPLACE FUNCTION person_exists(person_id in
PERSON.PERSON_ID%type)
    RETURN boolean
AS
    exist number;
BEGIN
    select case
        when exists(select * from PERSON P where P.PERSON_ID =
person_exists.person_id) then 1
        else 0
        end
    into exist
    from dual;
    if exist = 1 then
        return true;
    else
        return false;
    end if;
END;

CREATE OR REPLACE FUNCTION person_reservations(person_id number)
    RETURN person_reservation_table
AS
    result person_reservation_table;
BEGIN
    if not person_exists(person_reservations.person_id) then
        raise_application_error(-20001, 'person with id ' || person_id || '
not found');
    end if;

    select person_reservation(R.RESERVATION_ID, R.TRIP_NAME, R.COUNTRY,
R.TRIP_DATE, R.STATUS) bulk collect
    into result
    from RESERVATIONS R
    where R.PERSON_ID = person_reservations.person_id;
    return result;
END;

```

```

✓ select *
  from person_reservations( person_id: 4);

```

person\_reservations

Output person_reservations					
	RESERVATION_ID	TRIP_NAME	COUNTRY	TRIP_DATE	STATUS
1	4	Piękny Kraków	Polska	2023-07-03	C
2	7	Znów do Francji	Francja	2023-05-01	P
3	81	Heł	Polska	2023-05-01	N

/\*\*\*\*\*

```

create or replace type available_trips as OBJECT
(
    country          varchar2(50),
    trip_date        date,
    trip_name        varchar2(100),
    no_places        number,
    no_available_places number
);

create or replace type available_trips_table is table of available_trips;


/* named with f_... because there already exists a view with the same name
*/

CREATE OR REPLACE FUNCTION f_available_trips(country
COUNTRY.COUNTRY_NAME%type, date_from date, date_to date)
RETURN available_trips_table
AS
    result available_trips_table;
BEGIN
    if not date_from < date_to then
        raise_application_error(-20001, 'ending date cannot be before
starting date');
    end if;

    select available_trips(AvTr.country,
        AvTr.trip_date,
        AvTr.trip_name,
        AvTr.no_places,
        AvTr.num_of_available_places) bulk collect
    into result
    from AVAILABLETRIPS AvTr
    where date_from < TRIP_DATE
        and TRIP_DATE < date_to
        and AvTr.COUNTRY = f_available_trips.country;
    return result;
END;

```

Istniejące wycieczki:

✓  **Select \***  
 from TRIPS;

TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES
1	1 Francja	2022-09-12	Wycieczka do Paryża	3	2
2	2 Polska	2023-07-03	Piękny Kraków	5	2
3	3 Francja	2023-05-01	Znów do Francji	2	0
4	4 Polska	2023-05-01	Hel	7	4
5	5 Hiszpania	2023-02-02	Czemu by nie?	5	5

Wynik pierwszego zapytania:



✓ `select *`  
`from f_available_trips('Francja', TO_DATE('01-01-2023', 'DD-MM-YYYY'), TO_DATE('01-01-2024', 'DD-MM-YYYY'));`

Output f\_available\_trips

0 rows

Country Trip\_Date Trip\_Name No\_Places No\_Available\_Places

Komentarz to wyniku: Żadna wycieczka się nie wyświetliła, gdyż pierwsza nie mieści się w przedziale czasowym, a na drugą nie ma już miejsc

Wynik drugiego zapytania:

✓ `select *`  
`from f_available_trips('Polska', TO_DATE('01-06-2023', 'DD-MM-YYYY'), TO_DATE('02-09-2024', 'DD-MM-YYYY'));`

Output f\_available\_trips

1 row

Country	Trip_Date	Trip_Name	No_Places	No_Available_Places
1 Polska	2023-07-03	Piękny Kraków	5	2

Komentarz do wyniku: Pojawiła się tylko wycieczka o trip\_id 2, gdyż mieści się ona w przedziale czasowym

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

- AddReservation(trip\_id, person\_id), procedura powinna kontrolować czy wycieczka jeszcze się nie odbyła, i czy są wolne miejsca
- ModifyReservationStatus(reservation\_id, status), procedura kontrolować czy możliwa jest zmiana statusu, np. zmiana statusu już anulowanej wycieczki (przywrócenie do stanu aktywnego nie zawsze jest możliwa – może już nie być miejsc)
- ModifyNoPlaces(trip\_id, no\_places), nie wszystkie zmiany liczby miejsc są dozwolone, nie można zmniejszyć liczby miejsc na wartość poniżej liczby zarezerwowanych miejsc

Należy rozważyć użycie transakcji

Należy zwrócić uwagę na kontrolę parametrów (np. jeśli parametrem jest trip\_id to należy sprawdzić czy taka wycieczka istnieje, jeśli robimy rezerwację to należy sprawdzać czy są wolne miejsca itp..)

### ### Komentarz

Dodałem atrybuty trip\_id w widokach Trips oraz AvailableTrips, aby można było z nich skorzystać w procedurze add\_reservation. Stworzyłem też parę funkcji pomocniczych.

### ### Kod

```
CREATE OR REPLACE FUNCTION trip_is_available(trip_id TRIP.trip_id%type)
RETURN boolean
AS
```

```

        exist number;
BEGIN
    select case
        when exists(select * from AVAILABLETRIPS AvTr where
AvTr.TRIP_ID = trip_is_available.trip_id)
            then 1
            else 0
            end
    into exist
    from dual;
    if exist = 1 then
        return true;
    else
        return false;
    end if;
END;

CREATE OR REPLACE FUNCTION person_has_trip_reservation(person_id
PERSON.PERSON_ID%type, trip_id TRIP.trip_id%type)
    RETURN boolean
AS
    exist number;
BEGIN
    select case
        when exists(select *
                    from RESERVATION R
                    where R.TRIP_ID =
person_has_trip_reservation.trip_id
                    and R.PERSON_ID =
person_has_trip_reservation.person_id)
            then 1
            else 0
            end
    into exist
    from dual;
    if exist = 1 then
        return true;
    else
        return false;
    end if;
END;

create or replace procedure add_reservation(trip_id TRIP.TRIP_ID%type,
person_id PERSON.PERSON_ID%type)
    is
    reservation_id RESERVATION.RESERVATION_ID%type;
begin
    if not trip_exists(trip_id) then
        raise_application_error(-20001, 'trip with id ' || trip_id || '
doesn't exist');
    elsif not PERSON_EXISTS(person_id) then
        raise_application_error(-20002, 'person wit id ' || person_id || '
doesn't exist');
    elsif not trip_is_available(trip_id) then
        raise_application_error(-20003, 'trip with id ' || trip_id || ' is
not available');
    elsif person_has_trip_reservation(person_id, trip_id) then
        raise_application_error(-20004,
            'person with id ' || person_id || ' already

```

```

has a reservation for trip with id ' ||
        trip_id);

    end if;

    insert into RESERVATION (TRIP_ID, PERSON_ID, STATUS)
    VALUES (TRIP_ID, PERSON_ID, 'N') returning RESERVATION_ID into
RESERVATION_ID;

    add_log(reservation_id, 'N');
end;

```

Pokazuję wyniki:

Available trips:

```

select *
from AVAILBLETRIPS;

```

reservation\_exists()

TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES
1	2 Polska	2023-07-03	Piękny Kraków	5	2
2	4 Polska	2023-05-01	Heł	7	4

Próba dodania rezerwacji na niedostępną wycieczkę:

```

68
69 declare
70     TRIP_ID  TRIP.TRIP_ID %TYPE    := 1;
71     PERSON_ID PERSON.PERSON_ID %TYPE := 4;
72 begin
73     BD_409890.ADD_RESERVATION(
74         TRIP_ID => TRIP_ID,
75         PERSON_ID => PERSON_ID
76     );
77 end;
78

```

```

[72000][20003]
ORA-20003: trip with id 1 is not available
ORA-06512: przy "BD_409890.ADD_RESERVATION", linia 10
ORA-06512: przy linia 5
Position: 0

```

Rezerwacje:

```
select *  
from RESERVATION;
```

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	1	1	1	C
2	2	1	2	N
3	3	2	1	C
4	4	2	4	C
5	5	2	7	N
6	6	2	9	P
7	22	2	6	N
8	7	3	4	P
9	41	3	8	N
10	42	3	7	C
11	8	4	6	P
12	9	4	9	C
13	10	4	3	N
14	81	4	4	N

Próba dodania drugiej rezerwacji danej osoby na tę samą wycieczkę:

```
72 declare  
73     TRIP_ID  TRIP.TRIP_ID %TYPE      := 2;  
74     PERSON_ID PERSON.PERSON_ID %TYPE := 4;  
75 begin  
76     BD_409890.ADD_RESERVATION(  
77         TRIP_ID => TRIP_ID,  
78         PERSON_ID => PERSON_ID  
79     );  
80 end;
```

[72000][20004]

ORA-20004: person with id 4 already has a reservation for trip with id 2

ORA-06512: przy "BD\_409890.ADD\_RESERVATION", linia 12

ORA-06512: przy linia 5

Position: 0

Poprawne wstawienie:

```
✓ declare
    TRIP_ID    TRIP.TRIP_ID %TYPE    := 2;
    PERSON_ID  PERSON.PERSON_ID %TYPE := 5;
begin
    BD_409890.ADD_RESERVATION(
        TRIP_ID => TRIP_ID,
        PERSON_ID => PERSON_ID
    );
end;
```

Efekt:

```
✓ select *
from RESERVATION;
```

Output BD_409890.RESERVATION				
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	1	1	1	C
2	2	1	2	N
3	3	2	1	C
4	4	2	4	C
5	5	2	7	N
6	6	2	9	P
7	22	2	6	N
8	101	2	5	N
9	7	3	4	P
10	41	3	8	N
11	42	3	7	C
12	8	4	6	P
13	9	4	9	C
14	10	4	3	N
15	81	4	4	N

```
CREATE OR REPLACE FUNCTION reservation_exists(reservation_id
RESERVATION.RESERVATION_ID%type)
RETURN boolean
AS
    exist number;
BEGIN
    select case
```

```

        when exists(select *
                    from RESERVATION R
                    where R.RESERVATION_ID =
reservation_exists.reservation_id) then 1
        else 0
        end
    into exist
    from dual;
    if exist = 1 then
        return true;
    else
        return false;
    end if;
END;

CREATE OR REPLACE PROCEDURE modify_reservation_status(reservation_id
RESERVATION.TRIP_ID%type,
                                                    status
RESERVATION.STATUS%type)
is
    current_status RESERVATION.STATUS%type;
    trip_id        RESERVATION.TRIP_ID%type;
begin
    if not reservation_exists(reservation_id) then
        raise_application_error(-20001, 'reservation with id ' ||
reservation_id || ' doesn't exist');
    end if;

    select R.STATUS, R.TRIP_ID
    into current_status, trip_id
    from RESERVATION R
    where R.RESERVATION_ID = modify_reservation_status.reservation_id;

    if current_status = status then
        return;
    elsif current_status = 'N' or
        current_status = 'P' or
        (current_status = 'C' and trip_is_available(trip_id)) then
        update RESERVATION R
        set R.STATUS = modify_reservation_status.status
        where R.RESERVATION_ID = modify_reservation_status.RESERVATION_ID;
    else
        raise_application_error(-20005, 'cannot change status of
reservation with id ' || reservation_id ||
', no free places on the trip');
    end if;
end;

```

Dostępne wycieczki przed:

```

select *
from AVAILBLETRIPS;

```

Output BD_409890.AVAILBLETRIPS						
TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES	
1	2 Polska	2023-07-03	Piękny Kraków	5		1
2	4 Polska	2023-05-01	Hel	7		4

Wszystkie rezerwacje przed:

```
select *  
from RESERVATION;
```

Output BD_409890.RESERVATION				
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	1	1	1	C
2	2	1	2	N
3	3	2	1	C
4	4	2	4	C
5	5	2	7	N
6	6	2	9	P
7	7	3	4	P
8	8	4	6	P
9	9	4	9	C
10	10	4	3	N
11	41	3	8	N
12	22	2	6	N
13	42	3	7	C
14	81	4	4	N
15	101	2	5	N

Próba zmiany statusu wycieczki z 'C' na 'P' w wycieczce bez wolnych miejsc lub wycieczce, która już się odbyła:

```
124  
125 declare  
126     RESERVATION_ID RESERVATION.TRIP_ID %TYPE := 1;  
127     STATUS          RESERVATION.STATUS %TYPE := 'P';  
128 begin  
129     BD_409890.MODIFY_RESERVATION_STATUS(  
130         RESERVATION_ID => RESERVATION_ID,  
131         STATUS => STATUS  
132     );  
133 end;  
134
```

[72000][20005]

ORA-20005: cannot change status of reservation with id 1 , no free places on the trip

ORA-06512: przy "BD\_409890.MODIFY\_RESERVATION\_STATUS", linia 27

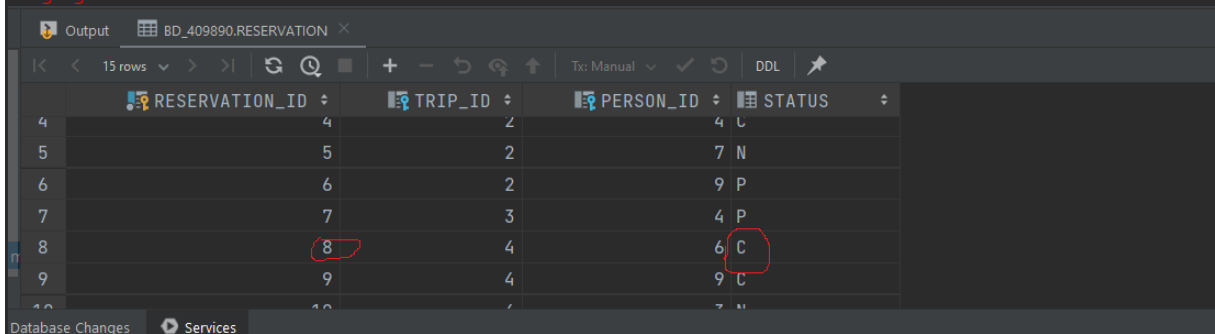
ORA-06512: przy linia 5

Position: 0

Poprawna zmiana:

```
declare
    RESERVATION_ID RESERVATION.TRIP_ID %TYPE := 8;
    STATUS          RESERVATION.STATUS %TYPE := 'C';
begin
    BD_409890.MODIFY_RESERVATION_STATUS(
        RESERVATION_ID => RESERVATION_ID,
        STATUS => STATUS
    );
end;
```

I jej efekt:



	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
4	4	2	4	L
5	5	2	7	N
6	6	2	9	P
7	7	3	4	P
8	8	4	6	C
9	9	4	9	C

```
create or replace procedure modify_no_places(trip_id
TRIP.MAX_NO_PLACES%type, no_places number)
is
    curr_no_places          number;
    curr_num_of_available_places number;
begin
    if not TRIP_EXISTS(trip_id) then
        raise_application_error(-20001, 'trip with id ' || trip_id || '
doesn't exist');
    end if;

    select T.NO_PLACES, NUM_OF_AVAILABLE_PLACES
    into curr_no_places, curr_num_of_available_places
    from TRIPS T
    where T.TRIP_ID = modify_no_places.trip_id;

    /* we can always increase number of places on a trip */
    /* we can lower number of places only by up to the number of currently
available places */
    if no_places > curr_no_places or curr_no_places - no_places <=
curr_num_of_available_places then
        update TRIP
        set MAX_NO_PLACES = no_places
        where TRIP_ID = modify_no_places.TRIP_ID;
    else
        raise_application_error(-20002,
                                'cannot set number of available places to '
|| no_places || ' for trip with id ' ||
                                trip_id ||
                                '; otherwise there would be more
reservations than places');
    end if;
end;
```



Wycieczki przed:

```
select *  
from TRIPS;  
modify_no_places()
```

	TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES
1	1	Francja	2022-09-12	Wycieczka do Paryża	3	2
2	2	Polska	2023-07-03	Piękny Kraków	5	2
3	3	Francja	2023-05-01	Znów do Francji	2	0
4	4	Polska	2023-05-01	Heł	7	5
5	5	Hiszpania	2023-02-02	Czemu by nie?	5	5

Zmiana liczby miejsc w wycieczce o trip\_id 2:

```
declare  
    TRIP_ID    TRIP.MAX_NO_PLACES %TYPE := 2;  
    NO_PLACES  number                := 3;  
  
begin  
    BD_409890.MODIFY_NO_PLACES(  
        TRIP_ID => TRIP_ID,  
        NO_PLACES => NO_PLACES  
    );  
end;
```

Efekt:

	TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES
1	1	Francja	2022-09-12	Wycieczka do Paryża	3	2
2	2	Polska	2023-07-03	Piękny Kraków	3	0
3	3	Francja	2023-05-01	Znów do Francji	2	0
4	4	Polska	2023-05-01	Heł	7	5
5	5	Hiszpania	2023-02-02	Czemu by nie?	5	5

Wywołanie niepoprawne:

```
151 declare  
152     TRIP_ID    TRIP.MAX_NO_PLACES %TYPE := 2;  
153     NO_PLACES  number                := 2;  
154  
155     begin  
156         BD_409890.MODIFY_NO_PLACES(  
157             TRIP_ID => TRIP_ID,  
158             NO_PLACES => NO_PLACES  
159         );  
160     end;
```

[72000][20002]

ORA-20002: cannot set number of available places to 2 for trip with id 2; otherwise there would be more reservations than places

ORA-06512: przy "BD\_409890.MODIFY\_NO\_PLACES", linia 23

ORA-06512: przy linia 5

Position: 0

6. Dodajemy tabelę dziennikującą zmiany statusu rezerwacji

Log(id, reservation\_id, date, status)

Należy zmienić warstwę procedur modyfikujących dane tak aby dopisywały informację do dziennika

### ### Kod

Kod tworzący tablicę dziennikową:

```
create table log
(
    log_id int generated always as identity not null,
    reservation_id int not null,
    log_date date not null,
    status char(1),
    constraint log_pk primary key ( log_id ) enable
);

alter table log
add constraint log_chk1 check
(status in ('N','P','C')) enable;

alter table log
add constraint log_fk1 foreign key
( reservation_id ) references reservation ( reservation_id ) enable;
```

Funkcja dodająca wpis to tabeli Log

```
create or replace procedure add_log(reservation_id
RESERVATION.RESERVATION_ID%type, status RESERVATION.STATUS%type)
is
begin
    if not reservation_exists(reservation_id) then
        raise_application_error(-20006, 'reservation with id ' ||
reservation_id || ' doesn't exist');
    end if;

    insert into LOG (RESERVATION_ID, LOG_DATE, STATUS)
VALUES (add_log.reservation_id, current_date, add_log.status);
end;
```

Zmodyfikowany / dodany kod poprzednio napisanych procedur:

w add\_reservation:

```
insert into RESERVATION (TRIP_ID, PERSON_ID, STATUS)
VALUES (TRIP_ID, PERSON_ID, 'N') returning RESERVATION_ID into
RESERVATION_ID;

add_log(reservation_id, 'N');
```

w modify\_reservation\_status:

```
update RESERVATION R
set R.STATUS = modify_reservation_status.status
where R.RESERVATION_ID = modify_reservation_status.RESERVATION_ID;

add_log(modify_reservation_status.reservation_id,
modify_reservation_status.status);
```

Tabela Log po paru wywołanych procedurach:

Stan przed:

```
select *  
from LOG;
```

Output BD\_409890.LOG

	LOG_ID	RESERVATION_ID	LOG_DATE	STATUS
1	1	3	2023-03-22 15:45:05	N
2	2	3	2023-03-22 15:45:17	C
3	21	81	2023-03-22 17:06:23	N
4	41	101	2023-03-22 17:37:42	N
5	61	8	2023-03-22 17:50:35	C
6	62	101	2023-03-22 17:52:00	C

Wywołania:

```
declare  
    TRIP_ID    TRIP.TRIP_ID %TYPE    := 4;  
    PERSON_ID  PERSON.PERSON_ID %TYPE := 10;  
begin  
    BD_409890.ADD_RESERVATION(  
        TRIP_ID => TRIP_ID,  
        PERSON_ID => PERSON_ID  
    );  
end;
```

```
declare  
    RESERVATION_ID RESERVATION.TRIP_ID %TYPE := 121;  
    STATUS          RESERVATION.STATUS %TYPE := 'C';  
begin  
    BD_409890.MODIFY_RESERVATION_STATUS(  
        RESERVATION_ID => RESERVATION_ID,  
        STATUS => STATUS  
    );  
end;
```

Stan po:

	LOG_ID	RESERVATION_ID	LOG_DATE	STATUS
1	1	3	2023-03-22 15:45:05	N
2	2	3	2023-03-22 15:45:17	C
3	21	81	2023-03-22 17:06:23	N
4	41	101	2023-03-22 17:37:42	N
5	61	8	2023-03-22 17:50:35	C
6	62	101	2023-03-22 17:52:00	C
7	81	121	2023-03-22 18:29:19	N
8	82	121	2023-03-22 18:30:00	C

## 7. Zmiana strategii zapisywania do dziennika rezerwacji. Realizacja przy pomocy triggerów

Należy wprowadzić zmianę, która spowoduje, że zapis do dziennika rezerwacji będzie realizowany przy pomocy triggerów

- trigger obsługujący dodanie rezerwacji
- trigger obsługujący zmianę statusu
- trigger zabraniający usunięcia rezerwacji

Oczywiście po wprowadzeniu tej zmiany należy uaktualnić procedury modyfikujące dane. Najlepiej to zrobić tworząc nowe wersje (np. z sufiksem 2)

### ### Komentarz

Procedury `add_reservation_2` i `modify_reservation_status_2` różnią się od swoich poprzedników tylko tym, że nie wywołują w swoim ciele procedury `add_log`. Zadanie to przejęły triggerzy.

### ### Kod

```
create or replace trigger reservation_added
  after insert or update
  on RESERVATION
  for each row
begin
  insert into LOG (RESERVATION_ID, LOG_DATE, STATUS)
  VALUES (:NEW.reservation_id, current_date, :NEW.status);
end;

create or replace trigger reservation_removed
  before delete
  on RESERVATION
begin
  raise_application_error(-20001, 'you cannot completely remove any
reservation from database;' ||
                                'if you want to cancel a reservation,
change it''s status to "C" (cancelled)');
end;
```

Wywołanie:

```
declare
    TRIP_ID    TRIP.TRIP_ID %TYPE    := 4;
    PERSON_ID  PERSON.PERSON_ID %TYPE := 5;
begin
    BD_409890.ADD_RESERVATION_2(
        TRIP_ID => TRIP_ID,
        PERSON_ID => PERSON_ID
    );
end;
```

Efekt:

	LOG_ID	RESERVATION_ID	LOG_DATE	STATUS
8	82	121	2023-03-22 18:30:00	C
9	83	122	2023-03-22 18:39:14	N
10	84	123	2023-03-22 18:41:59	N

Wywołanie:

```
declare
    RESERVATION_ID RESERVATION.TRIP_ID %TYPE := 123;
    STATUS          RESERVATION.STATUS %TYPE := 'C';
begin
    BD_409890.MODIFY_RESERVATION_STATUS_2(
        RESERVATION_ID => RESERVATION_ID,
        STATUS => STATUS
    );
end;
```

Efekt:

	LOG_ID	RESERVATION_ID	LOG_DATE	STATUS
9	83	122	2023-03-22 18:39:14	N
10	84	123	2023-03-22 18:41:59	N
11	85	123	2023-03-22 18:44:59	C

Próba usunięcia rezerwacji z tabeli:

```
19 delete from RESERVATION
20 where RESERVATION_ID = 5;
21
```

[72000][20001]

ORA-20001: you cannot completely remove any reservation from database;if you want to cancel a reservation, change it's status to "C" (cancelled)

ORA-06512: przy "BD\_409890.RESERVATION\_REMOVED", linia 2

ORA-04088: błąd w trakcie wykonywania wyzwalacza 'BD\_409890.RESERVATION\_REMOVED'

Position: 12

## 8. Zmiana strategii kontroli dostępności miejsc. Realizacja przy pomocy triggerów

Należy wprowadzić zmianę, która spowoduje, że zapis do dziennika rezerwacji będzie realizowany przy pomocy triggerów

trigger obsługujący dodanie rezerwacji

trigger obsługujący zmianę statusu

Oczywiście po wprowadzeniu tej zmiany należy uaktualnić procedury modyfikujące dane.

Najlepiej to zrobić tworząc nowe wersje (np. z sufiksem 3)

### Kod

# Triggery

```
create or replace trigger about_to_add_reservation
before insert
on RESERVATION
for each row
begin
    if not trip_is_available(:NEW.TRIP_ID) then
        raise_application_error(-20003, 'trip with id ' || :NEW.TRIP_ID ||
' is not available');
    elsif person_has_trip_reservation(:NEW.PERSON_ID, :NEW.TRIP_ID) then
        raise_application_error(-20004,
'person with id ' || :NEW.PERSON_ID || '
already has a reservation for trip with id ' ||
:NEW.TRIP_ID);
    end if;
end;

create or replace trigger about_to_modify_reservation
before update
on RESERVATION
for each row
begin
    /* cannot change reservation id, trip id nor person id */
    if :OLD.RESERVATION_ID != :NEW.RESERVATION_ID or
:OLD.TRIP_ID != :NEW.TRIP_ID or
:OLD.PERSON_ID != :NEW.PERSON_ID then
        raise_application_error(-20002, 'you cannot change reservation id,
trip id nor person id of any reservation');
    end if;

    if :OLD.STATUS = 'C' and not trip_is_available(:OLD.TRIP_ID) then
        raise_application_error(-20005, 'cannot change status of
reservation with id ' || :NEW.RESERVATION_ID ||
', no free places on the trip');
    end if;
end;
```

# Procedury w wersji nr. 3

```
/* These procedures are modified versions of previous procedures_2
They work together correctly with triggers from triggers_3 */

create or replace procedure add_reservation_3(trip_id TRIP.TRIP_ID%type,
person_id PERSON.PERSON_ID%type)
```

```

is
begin
    if not trip_exists(trip_id) then
        raise_application_error(-20001, 'trip with id ' || trip_id || '
doesn't exist');
    elsif not PERSON_EXISTS(person_id) then
        raise_application_error(-20002, 'person wit id ' || person_id || '
doesn't exist');
    end if;

    insert into RESERVATION (TRIP_ID, PERSON_ID, STATUS)
    VALUES (TRIP_ID, PERSON_ID, 'N');
end;

```

Przed:

✓ Select \*  
from AVAILBLETRIPS;

Scratches and Consoles

Output BD\_409890.AVAILABLETRIPS

	TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES
1	2	Polska	2023-07-03	Piękny Kraków	6	3
2	3	Francja	2023-05-01	Znów do Francji	5	3
3	4	Polska	2023-05-01	Heł	7	4

Wykonanie:

```

declare
    TRIP_ID TRIP.TRIP_ID %TYPE := 4;
    PERSON_ID PERSON.PERSON_ID %TYPE := 7;
begin
    BD_409890.ADD_RESERVATION_3(
        TRIP_ID => TRIP_ID,
        PERSON_ID => PERSON_ID
    );
end;

```

Efekt w Reservation:

Output BD\_409890.RESERVATION

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
15	101	2	5	C
16	121	4	10	C
17	122	4	8	N
18	123	4	5	C
19	126	4	7	N

Efekt w Log:

Output BD_409890.LOG X					
12 rows					
	LOG_ID	RESERVATION_ID	LOG_DATE	STATUS	
8	82	121	2023-03-22 18:30:00	C	
9	83	122	2023-03-22 18:39:14	N	
10	84	123	2023-03-22 18:41:59	N	
11	85	123	2023-03-22 18:44:59	C	
12	86	126	2023-03-22 20:46:02	N	

```
CREATE OR REPLACE PROCEDURE modify_reservation_status_3(reservation_id
RESERVATION.TRIP_ID%type,
                                status
RESERVATION.STATUS%type)
is
begin
    if not reservation_exists(reservation_id) then
        raise_application_error(-20001, 'reservation with id ' ||
reservation_id || ' doesn't exist');
    end if;

    update RESERVATION R
    set R.STATUS = modify_reservation_status_3.status
    where R.RESERVATION_ID = modify_reservation_status_3.RESERVATION_ID;
end;
```

Wykonanie:

```
declare
    RESERVATION_ID RESERVATION.TRIP_ID %TYPE := 126;
    STATUS          RESERVATION.STATUS %TYPE := 'P';
begin
    BD_409890.MODIFY_RESERVATION_STATUS_3(
        RESERVATION_ID => RESERVATION_ID,
        STATUS => STATUS
    );
end;
```

Efekt w Reservation:



Output BD_409890.RESERVATION					
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS	
15	101	2	5	C	
16	121	4	10	C	
17	122	4	8	N	
18	123	4	5	C	
19	126	4	7	P	

Efekt w Log:

Output BD_409890.LOG					
	LOG_ID	RESERVATION_ID	LOG_DATE	STATUS	
11	85	123	2023-03-22 18:44:59	C	
12	86	126	2023-03-22 20:46:02	N	
13	87	126	2023-03-22 20:50:29	P	

9. Zmiana struktury bazy danych, w tabeli wycieczki dodajemy redundantne pole `no_available_places`

Obsługa redundantnego pola (kontrola liczby dostępnych miejsc) w procedurach

Należy zmodyfikować zestaw widoków. Proponuję dodać kolejne widoki (np. z sufiksem 4), które pobierają informację o wolnych miejscach z nowo dodanego pola.

Należy napisać procedurę przelicz która zaktualizuje wartość liczby wolnych miejsc dla już istniejących danych

Należy zmodyfikować warstwę procedur/funkcji pobierających dane, podobnie jak w przypadku widoków.

Należy zmodyfikować procedury wprowadzające dane tak aby korzystały/aktualizowały pole `no_available_places` w tabeli wycieczki

Najlepiej to zrobić tworząc nowe wersje (np. z sufiksem 4)

### ### Komentarz

Niestety, przez moje niezrozumienie polecenia, w celu przebudowy struktury bazy danych, usunąłem wszystkie dotychczasowe dane. Gdy zdałem sobie sprawę, że powinienem był dodać kolumnę `no_available_places` a następnie wpisać do niej procedurą liczbę dostępnych obecnie miejsc, było już po moim błędzie. Z tego też powodu wypełniłem jeszcze raz bazę różnymi danymi, a następnie (tym razem poprawnie) dodałem żadaną kolumnę i wpisałem do niej dane.

### Kod

# Modyfikacja tablicy:

```
alter table trip
  add no_available_places int;

alter table trip
  add constraint no_available_places_check1 check
    ( no_available_places >= 0 ) enable;
```

# Wpisanie wartości do no\_available\_places

```
create procedure calculate_no_available_places
  is
begin
  update TRIP T
  set T.no_available_places = NVL((select NUM_OF_AVAILABLE_PLACES
    from AVAILABLETRIPS AvTr
    where AvTr.TRIP_ID = T.TRIP_ID),
T.MAX_NO_PLACES);
end;
```

Rezerwacje:

Output BD_409890.RESERVATION					
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS	
1	3	2	1	C	
2	4	2	4	C	
3	5	2	7	N	
4	6	2	9	P	
5	7	3	4	C	
6	8	4	6	P	
7	9	4	9	C	
8	10	4	3	N	
9	11	1	1	P	
10	12	1	2	N	
11	41	3	2	N	

Przed:

```
19 select *
20 from TRIP;
21
```

OutputBD\_409890.TRIP

5 rows

</

Po:

```

27 ✓ select *
28 from TRIPS;
29

```

TRIP_ID	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NUM_OF_AVAILABLE_PLACES
1	Francja	2022-09-12	Wycieczka do Paryża	6	4
2	Polska	2023-07-03	Piękny Kraków	7	5
3	Francja	2023-05-01	Znów do Francji	5	4
4	Polska	2023-05-01	Hel	7	5
5	Hiszpania	2023-02-02	Czemu by nie?	5	5

## # Widoki

# Komentarz: Widok Trips\_4 uznałem za niepotrzebny, gdyż w obecnej sytuacji pokrywa się on z samą tabelą Trip

```

CREATE OR REPLACE VIEW Reservations_4 (trip_id, country, trip_date,
trip_name, person_id, firstname, lastname, reservation_id, status) AS
select T.TRIP_ID, T.COUNTRY, T.TRIP_DATE, T.TRIP_NAME, P.PERSON_ID,
P.FIRSTNAME, P.LASTNAME, R.RESERVATION_ID, R.STATUS
from RESERVATION R
    join TRIP T on R.TRIP_ID = T.TRIP_ID
    join PERSON P on P.PERSON_ID = R.PERSON_ID;

CREATE OR REPLACE VIEW AvailableTrips_4 (trip_id, country, trip_date,
trip_name, no_places, num_of_available_places) AS
select TRIP_ID, COUNTRY, TRIP_DATE, TRIP_NAME, MAX_NO_PLACES,
NO_AVAILABLE_PLACES
from Trip
where NO_AVAILABLE_PLACES > 0
    and current_date < TRIP_DATE;

```

## # Funkcje zwracające dane

# Zmienił się kod jedynie poniższej funkcji (zmiana tablicy źródłowej)

```

CREATE OR REPLACE FUNCTION f_available_trips_4(country
COUNTRY.COUNTRY_NAME%type, date_from date, date_to date)
RETURN available_trips_table
AS
    result available_trips_table;
BEGIN
    if not date_from < date_to then
        raise_application_error(-20001, 'ending date cannot be before
starting date');
    end if;

    select available_trips(T.country,
        T.trip_date,
        T.trip_name,
        T.max_no_places,
        T.no_available_places) bulk collect
    into result
    from TRIP T
    where date_from < TRIP_DATE
        and TRIP_DATE < date_to
        and T.COUNTRY = f_available_trips_4.country;
    return result;
END;

```

### Procedury modyfikujące dane:

```
create or replace procedure add_reservation_4(trip_id TRIP.TRIP_ID%type,
person_id PERSON.PERSON_ID%type)
is
begin
    if not trip_exists(trip_id) then
        raise_application_error(-20001, 'trip with id ' || trip_id || '
doesn''t exist');
    elsif not PERSON_EXISTS(person_id) then
        raise_application_error(-20002, 'person wit id ' || person_id || '
doesn''t exist');
    elsif not trip_is_available(trip_id) then
        raise_application_error(-20003, 'trip with id ' || trip_id || ' is
not available');
    elsif person_has_trip_reservation(person_id, trip_id) then
        raise_application_error(-20004,
            'person with id ' || person_id || ' already
has a reservation for trip with id ' ||
            trip_id);
    end if;

    insert into RESERVATION (TRIP_ID, PERSON_ID, STATUS)
    VALUES (TRIP_ID, PERSON_ID, 'N');

    update TRIP
    set NO_AVAILABLE_PLACES = NO_AVAILABLE_PLACES - 1
    where TRIP_ID = add_reservation_4.TRIP_ID;
end;

CREATE OR REPLACE PROCEDURE modify_reservation_status_4(reservation_id
RESERVATION.TRIP_ID%type,
status
RESERVATION.STATUS%type)
is
    current_status RESERVATION.STATUS%type;
    trip_id RESERVATION.TRIP_ID%type;
begin
    if not reservation_exists(reservation_id) then
        raise_application_error(-20001, 'reservation with id ' ||
reservation_id || ' doesn''t exist');
    end if;

    select R.STATUS, R.TRIP_ID
    into current_status, trip_id
    from RESERVATION R
    where R.RESERVATION_ID = modify_reservation_status_4.reservation_id;

    if current_status = status then
        return;

    elsif (current_status = 'N' or current_status = 'P') and status = 'C'
then
        update RESERVATION R
        set R.STATUS = modify_reservation_status_4.status
        where R.RESERVATION_ID =
modify_reservation_status_4.RESERVATION_ID;

        update TRIP
        set NO_AVAILABLE_PLACES = NO_AVAILABLE_PLACES + 1
        where TRIP_ID = modify_reservation_status_4.trip_id;
```

```

    elsif (current_status = 'N' or current_status = 'P') and (status = 'N'
or status = 'P') then
        update RESERVATION R
        set R.STATUS = modify_reservation_status_4.status
        where R.RESERVATION_ID =
modify_reservation_status_4.RESERVATION_ID;

        elsif current_status = 'C' and TRIP_IS_AVAILABLE(trip_id) then
            update RESERVATION R
            set R.STATUS = modify_reservation_status_4.status
            where R.RESERVATION_ID =
modify_reservation_status_4.RESERVATION_ID;

            update TRIP
            set NO_AVAILABLE_PLACES = NO_AVAILABLE_PLACES - 1
            where TRIP_ID = modify_reservation_status_4.trip_id;

        else
            raise_application_error(-20005, 'cannot change status of
reservation with id ' || reservation_id ||
', no free places on the trip');
        end if;
    end;

```

Będziemy modyfikować rezerwację nr. 41. Zmienimy jej status na 'C':

Output BD_409890.RESERVATION				
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
10	12	1	2	N
11	41	3	2	N
12	42	4	5	N

Przed:

Output BD_409890.TRIP					
	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	NO_AVAILABLE_PLACES
1	1	Wycieczka do Paryża	Francja	2022-09-12	6
2	2	Piękny Kraków	Polska	2023-07-03	5
3	3	Znów do Francji	Francja	2023-05-01	4
4	4	Heł	Polska	2023-05-01	4
5	5	Czemu by nie?	Hiszpania	2023-02-02	5

Output BD_409890.RESERVATION				
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
7	11	1	1	P
10	12	1	2	N
11	41	3	2	C
12	42	4	5	N

Po:

Output BD_409890.RESERVATION				
12 rows				
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
8	10	4	3	N
9	11	1	1	P
10	12	1	2	N
11	41	3	2	P
12	42	4	5	N

Output BD_409890.TRIP					
5 rows					
	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES
1	1	Wycieczka do Paryża	Francja	2022-09-12	6
2	2	Piękny Kraków	Polska	2023-07-03	7
3	3	Znów do Francji	Francja	2023-05-01	5
4	4	Heł	Polska	2023-05-01	7
5	5	Czemu by nie?	Hiszpania	2023-02-02	5

Z powrotem zmieniamy status na 'N':

Output BD_409890.TRIP					
5 rows					
	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES
1	1	Wycieczka do Paryża	Francja	2022-09-12	6
2	2	Piękny Kraków	Polska	2023-07-03	7
3	3	Znów do Francji	Francja	2023-05-01	5
4	4	Heł	Polska	2023-05-01	7
5	5	Czemu by nie?	Hiszpania	2023-02-02	5

Wszystko zarejestrowała tablica Log:

Output BD_409890.LOG				
18 rows				
	LOG_ID	RESERVATION_ID	LOG_DATE	STATUS
15	16	41	2023-03-23 13:49:38	N
16	17	42	2023-03-23 15:21:24	N
17	18	41	2023-03-23 15:36:52	C
18	19	41	2023-03-23 15:37:43	P

```
create or replace procedure modify_no_places_4(trip_id
TRIP.MAX_NO_PLACES%type, no_places number)
is
    curr_no_places          number;
    curr_num_of_available_places number;
begin
    if not TRIP_EXISTS(trip_id) then
        raise_application_error(-20001, 'trip with id ' || trip_id || '
doesn't exist');
    end if;

    select T.MAX_NO_PLACES, NO_AVAILABLE_PLACES
    into curr no places, curr num of available places
```

```

from TRIP T
where T.TRIP_ID = modify_no_places_4.trip_id;

/* we can always increase number of places on a trip */
/* we can lower number of places only by up to the number of currently
available places */
if no_places > curr_no_places or curr_no_places - no_places <=
curr_num_of_available_places then
    update TRIP
    set MAX_NO_PLACES = no_places, NO_AVAILABLE_PLACES =
NO_AVAILABLE_PLACES + no_places - curr_no_places
    where TRIP_ID = modify_no_places_4.TRIP_ID;
else
    raise_application_error(-20002,
        'cannot set number of available places to '
|| no_places || ' for trip with id ' ||
        trip_id ||
        '; otherwise there would be more
reservations than places');
end if;
end;

```

10. Zmiana strategii obsługi redundantnego pola no\_available\_places, realizacja przy pomocy triggerów

trigger obsługujący dodanie rezerwacji

trigger obsługujący zmianę statusu

trigger obsługujący zmianę liczby miejsc na poziomie wycieczki

Oczywiście po wprowadzeniu tej zmiany należy uaktualnić procedury modyfikujące dane.

Najlepiej to zrobić tworząc nowe wersje (np. z sufiksem 5)

### Kod

# Triggery

```

create or replace trigger about_to_add_reservation_5
before insert
on RESERVATION
for each row
begin
    if not trip_is_available(:NEW.TRIP_ID) then
        raise_application_error(-20003, 'trip with id ' || :NEW.TRIP_ID ||
' is not available');
    elsif person_has_trip_reservation(:NEW.PERSON_ID, :NEW.TRIP_ID) then
        raise_application_error(-20004,
            'person with id ' || :NEW.PERSON_ID || '
already has a reservation for trip with id ' ||
            :NEW.TRIP_ID);
    end if;

    update TRIP
    set NO_AVAILABLE_PLACES = NO_AVAILABLE_PLACES - 1
    where TRIP_ID = :NEW.TRIP_ID;
end;

```

```

create or replace trigger about_to_modify_reservation_5
before update
on RESERVATION
for each row
begin
    /* cannot change reservation id, trip id nor person id */
    if :OLD.RESERVATION_ID != :NEW.RESERVATION_ID or
       :OLD.TRIP_ID != :NEW.TRIP_ID or
       :OLD.PERSON_ID != :NEW.PERSON_ID then
        raise_application_error(-20002, 'you cannot change reservation id,
trip id nor person id of any reservation');
    end if;

    if :OLD.STATUS = :NEW.STATUS then
        return;

    elsif (:OLD.STATUS = 'N' or :OLD.STATUS = 'P') and :NEW.STATUS = 'C'
then
        update TRIP
        set NO_AVAILABLE_PLACES = NO_AVAILABLE_PLACES + 1
        where TRIP_ID = :OLD.TRIP_ID;

    elsif :OLD.STATUS = 'C' and TRIP_IS_AVAILABLE(:OLD.TRIP_ID) then
        update TRIP
        set NO_AVAILABLE_PLACES = NO_AVAILABLE_PLACES - 1
        where TRIP_ID = :OLD.TRIP_ID;

    elsif :OLD.STATUS = 'C' and not trip_is_available(:OLD.TRIP_ID) then
        raise_application_error(-20005, 'cannot change status of
reservation with id ' || :NEW.RESERVATION_ID ||
        ', no free places on the trip');
    else
        raise_application_error(-20006, 'unknown error');
    end if;
end;

```

### # Procedury

```

create or replace procedure add_reservation_5(trip_id TRIP.TRIP_ID%type,
person_id PERSON.PERSON_ID%type)
is
begin
    if not trip_exists(trip_id) then
        raise_application_error(-20001, 'trip with id ' || trip_id || '
doesn't exist');
    elsif not PERSON_EXISTS(person_id) then
        raise_application_error(-20002, 'person wit id ' || person_id || '
doesn't exist');
    end if;

    insert into RESERVATION (TRIP_ID, PERSON_ID, STATUS)
    VALUES (TRIP_ID, PERSON_ID, 'N');
end;

```



Przed wstawieniem:

Output BD_409890.TRIP						
	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	1	Wycieczka do Paryza	Francja	2022-09-12	6	6
2	2	Piękny Kraków	Polska	2023-07-03	7	4
3	3	Znów do Francji	Francja	2023-05-01	5	4
4	4	Heł	Polska	2023-05-01	7	4
5	5	Czemu by nie?	Hiszpania	2023-02-02	5	5

Po wstawieniu:

Output BD_409890.TRIP						
	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	1	Wycieczka do Paryza	Francja	2022-09-12	6	6
2	2	Piękny Kraków	Polska	2023-07-03	7	3
3	3	Znów do Francji	Francja	2023-05-01	5	4
4	4	Heł	Polska	2023-05-01	7	4
5	5	Czemu by nie?	Hiszpania	2023-02-02	5	5

Wstawiona rezerwacja:

Output BD_409890.RESERVATION				
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
12	42	4	5	N
13	61	2	3	N
14	63	2	5	N

```
CREATE OR REPLACE PROCEDURE modify_reservation_status_5(reservation_id
RESERVATION.TRIP_ID%type,
                                status
RESERVATION.STATUS%type)
is
begin
    if not reservation_exists(reservation_id) then
        raise_application_error(-20001, 'reservation with id ' ||
reservation_id || ' doesn't exist');
    end if;

    update RESERVATION R
    set R.STATUS = modify_reservation_status_5.status
    where R.RESERVATION_ID = modify_reservation_status_5.RESERVATION_ID;
end;
```

Przed:

Output BD_409890.RESERVATION				
	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
12	42	4	5	N
13	61	2	3	N
14	63	2	5	N

Output BD\_409890.TRIP

	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	1	Wycieczka do Paryża	Francja	2022-09-12	6	6
2	2	Piękny Kraków	Polska	2023-07-03	7	3
3	3	Znów do Francji	Francja	2023-05-01	5	4
4	4	Heł	Polska	2023-05-01	7	4
5	5	Czemu by nie?	Hiszpania	2023-02-02	5	5

Po zmianie statusu na 'C':

Output BD\_409890.RESERVATION

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
12	42	4	5	N
13	61	2	3	N
14	63	2	5	C

## Uwagi

Należy przygotować raport z wykonania ćwiczenia. Raport powinien zawierać polecenia SQL (między innymi kod widoków, procedur, ...), wynik działania oraz krótki komentarz. Raport należy przesłać w formie pliku PDF.

W raporcie można posłużyć się zrzutami ekranów. Dodatkowo proszę załączyć kod zaimplementowanych widoków/procedur postaci pliku tekstowego (plik tekstowy z rozszerzeniem sql)

Proszę zwrócić uwagę na formatowanie kodu (struktura)

Punktacja za wykonanie pkt. I ćw (wersja dla SZBD Oracle) - max 2pkt