

# Laboratorium 1 : Oracle PL/SQL

Bartłomiej Kozera

## 1. Tabele

### 1.1. Tabela person

```
create table person
(
    person_id int generated always as identity not null,
    firstname varchar(50),
    lastname varchar(50),
    constraint person_pk primary key ( person_id ) enable
);
```

### 1.2. Tabela country

```
create table country
(
    country varchar(50),
    constraint country_pk primary key ( country ) enable
);
```

### 1.3. Tabela trip wraz z kluczem obcym country

```
create table trip
(
    trip_id int generated always as identity not null,
    trip_name varchar(100),
    country varchar(50),
    trip_date date,
    max_no_places int,
    constraint trip_pk primary key ( trip_id ) enable
);

alter table trip
add constraint trips_fk1 foreign key
( country ) references countries ( country ) enable;
```

#### 1.4. Tabela reservation wraz z kluczami obcymi i sprawdzaniem poprawności danych

```
create table reservation
(
  reservation_id int generated always as identity not null,
  trip_id int,
  person_id int,
  status char(1),
  constraint reservation_pk primary key ( reservation_id ) enable
);

alter table reservation
add constraint reservation_fk1 foreign key
( person_id ) references person ( person_id ) enable;

alter table reservation
add constraint reservation_fk2 foreign key
( trip_id ) references trip ( trip_id ) enable;

alter table reservation
add constraint reservation_chk1 check
(status in ('N','P','C')) enable;
```

## 2. Dane poszczególnych tabel

Wszystkie dane dodawane były do tabel za pomocą poleceni insert into [nazwa tabeli] (param\_1, param\_2, ..., param\_n) values (wartość\_1, wartość\_2, ..., wartość\_n).

### 2.1. Dane tabeli person

	PERSON_ID	FIRSTNAME	LASTNAME
1	1	Jan	Nowak
2	2	Jan	Kowalski
3	3	Jan	Nowakowski
4	4	Adam	Kowalski
5	5	Novak	Nowak
6	6	Piotr	Piotrowski
7	7	Robert	Lewandowski
8	8	Adam	Malysz
9	9	Wojciech	Kowalczyk
10	10	Stanisław	Wokułski

## 2.2. Dane tabeli country

	COUNTRY
1	Francja
2	Polska

## 2.3. Dane tabeli trip

	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES
1	1	Wycieczka do Paryża	Francja	2022-09-12	3
2	2	Piękny Kraków	Polska	2023-07-03	10
3	3	Znów do Francji	Francja	2023-05-01	15
4	4	Hel	Polska	2023-01-01	8

## 2.4. Dane tabeli reservation

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	1	1	1	P
2	2	1	2	C
3	3	2	1	P
4	4	2	4	N
5	5	2	4	P
6	6	3	7	C
7	7	3	8	N
8	8	3	2	P
9	9	4	8	C
10	10	4	9	P

# 3. Widoki

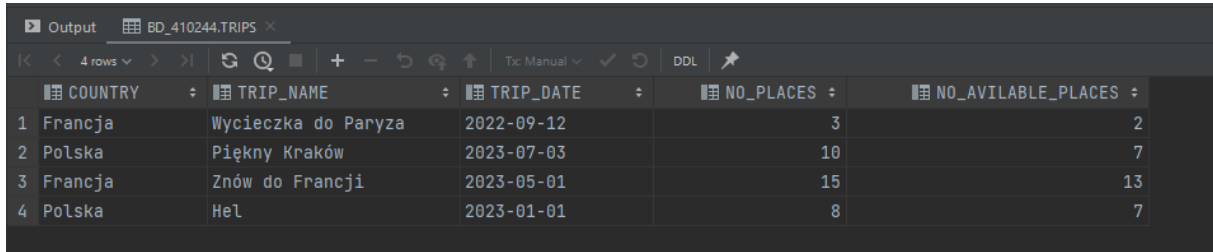
## 3.1. Widok Reservations

```
CREATE OR REPLACE VIEW Reservations AS
SELECT country, trip_name, trip_date, FIRSTNAME, LASTNAME, RESERVATION_ID, STATUS
FROM RESERVATION R
inner join TRIP T on T.TRIP_ID = R.TRIP_ID
inner join PERSON P on P.PERSON_ID = R.PERSON_ID
;
```

	COUNTRY	TRIP_NAME	TRIP_DATE	FIRSTNAME	LASTNAME	RESERVATION_ID	STATUS
1	Francja	Wycieczka do Paryża	2022-09-12	Jan	Nowak	1	P
2	Polska	Piękny Kraków	2023-07-03	Jan	Nowak	3	P
3	Francja	Wycieczka do Paryża	2022-09-12	Jan	Kowalski	2	C
4	Francja	Znów do Francji	2023-05-01	Jan	Kowalski	8	P
5	Polska	Piękny Kraków	2023-07-03	Adam	Kowalski	4	N
6	Polska	Piękny Kraków	2023-07-03	Adam	Kowalski	5	P
7	Francja	Znów do Francji	2023-05-01	Robert	Lewandowski	6	C
8	Francja	Znów do Francji	2023-05-01	Adam	Malysz	7	N
9	Polska	Hel	2023-01-01	Adam	Malysz	9	C
10	Polska	Hel	2023-01-01	Wojciech	Kowalczyk	10	P

### 3.2. Widok Trips

```
CREATE OR REPLACE VIEW Trips AS
SELECT T.country, T.trip_name, T.trip_date, MAX_NO_PLACES no_places,
       MAX_NO_PLACES - (select count(*) from RESERVATION where
                        RESERVATION.TRIIP_ID = T.TRIIP_ID and RESERVATION.STATUS != 'C') no_available_places
from TRIP T
```

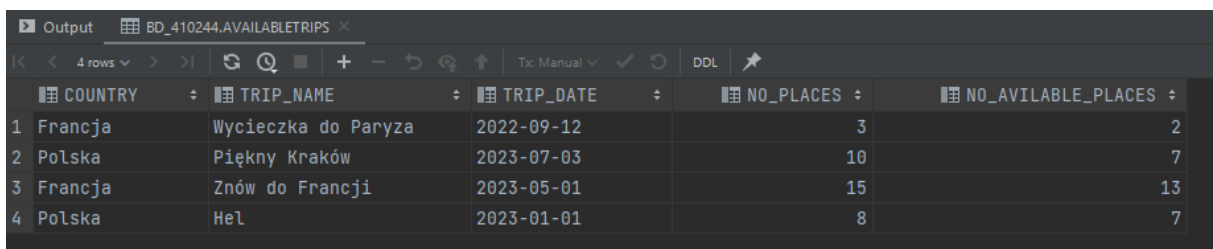


Output window showing the result of the Trips view query. The table has 5 columns: COUNTRY, TRIP\_NAME, TRIP\_DATE, NO\_PLACES, and NO\_AVAILABLE\_PLACES. There are 4 rows of data.

	COUNTRY	TRIP_NAME	TRIP_DATE	NO_PLACES	NO_AVAILABLE_PLACES
1	Francja	Wycieczka do Paryza	2022-09-12	3	2
2	Polska	Piękny Kraków	2023-07-03	10	7
3	Francja	Znów do Francji	2023-05-01	15	13
4	Polska	Hel	2023-01-01	8	7

### 3.3. Widok AvailableTrips

```
CREATE OR REPLACE VIEW AvailableTrips AS
SELECT country, trip_name, trip_date, no_places, no_available_places
FROM Trips
where no_available_places > 0
;
```



Output window showing the result of the AvailableTrips view query. The table has 5 columns: COUNTRY, TRIP\_NAME, TRIP\_DATE, NO\_PLACES, and NO\_AVAILABLE\_PLACES. There are 4 rows of data, identical to the Trips view.

	COUNTRY	TRIP_NAME	TRIP_DATE	NO_PLACES	NO_AVAILABLE_PLACES
1	Francja	Wycieczka do Paryza	2022-09-12	3	2
2	Polska	Piękny Kraków	2023-07-03	10	7
3	Francja	Znów do Francji	2023-05-01	15	13
4	Polska	Hel	2023-01-01	8	7

### 3.4. Wnioski

Tworzenie widoku przypomina jego konstrukcję w MsSQL, jedyną sporą różnicą konstrukcyjną którą zauważyłem jest sformułowanie 'create or replace' które pozwala nam na stworzenie lub jeśli taki widok istnieje, uaktualnienie go. Widoki z podpunktu 3.2. oraz 3.3. nie różnią się wyjściem ze względu na dane, nie ma wycieczki na której nie było by miejsc.

## 4. Procedury

### 4.1. Obiekty

Aby w PL/SQL stworzyć procedurę która zwraca nam tabelę z odpowiadającymi nam danymi, należy stworzyć 2 typy pól. Pierwszym z nich jest obiekt, który będzie zawierał odpowiednie pola, wraz z poprawnymi typami tych pól oraz tabele tychże obiektów. Kolejna różnica między tym językiem a MsSQL, gdzie były 2 typy procedur, jedne zwracały wartość, drugie tabele, tutaj procedury tylko zwracają wartość, stąd potrzeba tworzenie obiektów oraz ich tabel.

```

create or replace type trip_participant as OBJECT
(
    FIRSTNAME varchar(50),
    LASTNAME varchar(50),
    STATUS char(1),
    RESERVATION_ID int
);
create or replace type trip_participant_table is table of trip_participant;

create or replace type person_reservation as OBJECT
(
    country varchar(50),
    trip_name varchar(50),
    trip_date date,
    RESERVATION_ID int,
    STATUS char(1)
);
create or replace type person_reservation_table is table of person_reservation;

create or replace type available_trips as OBJECT
(
    country varchar(50),
    trip_name varchar(50),
    trip_date date,
    max_no_places int
);
create or replace type available_trips_table is table of available_trips;

```

#### 4.2. Funkcja TripParticipants

```

create or replace function TripParticipants(trip_id int)
return trip_participant_table
as
result trip_participant_table;
valid int;
begin
    select count(*) into valid
    from trip t
    where t.trip_id = TripParticipants.trip_id;
    if valid = 0 then
        raise_application_error(-20001, 'trip not found');
    end if;
    select trip_participant(rs.FIRSTNAME, rs.LASTNAME, rs.STATUS, rs.RESERVATION_ID) bulk collect
    into result
    from RESERVATIONS rs
    join RESERVATION r on r.RESERVATION_ID = rs.RESERVATION_ID
    where r.trip_id = TripParticipants.trip_id;
    return result;
end;

```

Po wykonaniu tego polecenia:

```
select * from TRIPPARTICIPANTS( TRIP_ID: 2)
```

Wynikami są:

	FIRSTNAME	LASTNAME	STATUS	RESERVATION_ID
1	Jan	Nowak	P	3
2	Adam	Kowalski	P	5
3	Adam	Kowalski	N	4

#### 4.3. Funkcja PersonReservation

```
create or replace function PersonReservations(person_id int)
    return person_reservation_table
as
    result person_reservation_table;
    valid int;
begin
    select count(*) into valid
        from PERSON p
        where p.PERSON_ID = PersonReservations.person_id;
    if valid = 0 then
        raise_application_error(-20001, 'person not found');
    end if;
    select person_reservation(rs.COUNTRY, rs.TRIP_NAME,
        rs.TRIP_DATE, rs.RESERVATION_ID, rs.STATUS) bulk collect
    into result
    from RESERVATIONS rs
    join RESERVATION r on r.RESERVATION_ID = rs.RESERVATION_ID
    join PERSON p on p.PERSON_ID = r.PERSON_ID
    where p.PERSON_ID = PersonReservations.person_id;
    return result;
end;
```

Po wykonaniu tego polecenia:

```
select * from PERSONRESERVATIONS( PERSON_ID: 2)
```

Wynikami są:

	COUNTRY	TRIP_NAME	TRIP_DATE	RESERVATION_ID	STATUS
1	Francja	Wycieczka do Paryża	2022-09-12	2	C
2	Francja	Znów do Francji	2023-05-01	8	P

#### 4.4. Funkcja FAvailableTrips

```
create or replace function FAvailableTrips(country varchar, date_from date, date_to date)
    return available_trips_table
as
    result available_trips_table;
    valid int;
begin
    if FAvailableTrips.date_to < FAvailableTrips.date_from then
        raise_application_error(-20001, 'date_to is earlier from date_from');
    end if;
    select count(*) into valid
        from COUNTRIES c
        where c.COUNTRY = FAvailableTrips.country;
    if valid = 0 then
        raise_application_error(-20001, 'country not found');
    end if;
    select available_trips(t.country, trip_name, trip_date, max_no_places) bulk collect
        into result
        from TRIP t
        where t.country = FAvailableTrips.country
            and trip_date between FAvailableTrips.date_from and FAvailableTrips.date_to;
    return result;
end;
```

Po wykonaniu tego polecenia:

```
select * from FAVAILABLETRIPS( COUNTRY: 'Francja', DATE_FROM: '2022-10-29', DATE_TO: '2023-06-02')
```

Wynikami są:

	COUNTRY	TRIP_NAME	TRIP_DATE	MAX_NO_PLACES
1	Francja	Znów do Francji	2023-05-01	15



## 5. Procedury modyfikujące

### 5.1. Procedura AddReservation

```
create or replace procedure AddReservation(p_trip_id int, p_pers_id int)
as
    is_person char(1);
    are_trip_ok char(1);
begin
    select 1 into is_person from PERSON where PERSON_ID = p_pers_id;
    select 1 into are_trip_ok from AVAILABLETRIPS
        where TRIP_ID = p_trip_id and TRIP_DATE > current_date;
    insert into RESERVATION(trip_id, person_id, status)
    values (p_trip_id, p_pers_id, 'N');
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'incorrect data');
end;
```

Po wykonania poniższego polecenia

```
begin
    AddReservation( p_trip_id: 2, p_pers_id: 2);
    commit;
end;
```

Wyniki prawidłowo dodają się do tabeli z rezerwacjami

41	2	2	N
----	---	---	---

### 5.2. Procedure modify\_reservation\_status

```
create or replace procedure modify_reservation_status(p_reservation_id int, p_status char)
as
    tmp char(1);
    av_places int;
begin
    select 1 into tmp from reservation where reservation_id = p_reservation_id;
    if p_status not in ('N', 'P', 'C') then
        raise_application_error(-20002, 'wrong status');
    end if;
    select NO_AVAILABLE_PLACES into av_places from TRIPS T join RESERVATION R on R.TRIP_ID = T.TRIP_ID
        where RESERVATION_ID = p_reservation_id;
    if p_status in ('N', 'P') and av_places = 0 then
        raise_application_error(-20002, 'not enough places');
    end if;
    update reservation
    set status = p_status
    where reservation_id = p_reservation_id;
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'reservation not found');
end;
```

1	1	1	N
---	---	---	---

```
begin
    modify_reservation_status( p_reservation_id: 1, p_status: 'P');
    commit;
end;
```

1	1	1	P
---	---	---	---

Jak widać procedura poprawnie modyfikuje status zamówienia

### 5.3. Procedura modify\_no\_places

```
create or replace procedure modify_no_places(p_trip_id int, no_places int)
as
    tmp char(1);
    min_places int;
begin
    select 1 into tmp from TRIP where TRIP_ID = p_trip_id;
    select T.no_places - NO_AVAILABLE_PLACES into min_places from TRIPS T where TRIP_ID = p_trip_id;
    if min_places > no_places then
        raise_application_error(-20002, 'not enough places');
    end if;
    update TRIP
    set MAX_NO_PLACES = no_places
    where TRIP_ID = p_trip_id;
end;
```

1 Wycieczka do Paryza	Francja	2022-09-12	3
-----------------------	---------	------------	---

Po wykonaniu poniższego fragmentu kodu.

```
begin
    modify_no_places( p_trip_id: 1, no_places: 4);
    commit;
end;
```

Wartość pola max\_no\_places zmienia się.

1 Wycieczka do Paryza	Francja	2022-09-12	4
-----------------------	---------	------------	---

### 5.4. Wnioski

Procedury praktycznie nie różniące się od tych w ms SQL, Możemy zauważyć ciekawy sposób na wywoływanie błędów jako brak wartości dla zadeklarowanych przez nas argumentów pomocniczych. Nie da się stworzyć argumentów w sekcji begin ... end.

## 6. Tabela log

Kod do stworzenia tej tabeli był podany w raz z materiałami do zajęć, nie będę go tutaj umieszczał.

## 6.1. Modyfikacja procedury AddReservation

```
create or replace procedure AddReservation(p_trip_id int, p_pers_id int)
as
    is_person char(1);
    are_trip_ok char(1);
    max_id int;
begin
    select 1 into is_person from PERSON where PERSON_ID = p_pers_id;
    select 1 into are_trip_ok from AVAILBLETRIPS
        where TRIP_ID = p_trip_id and TRIP_DATE > current_date;
    insert into RESERVATION(trip_id, person_id, status)
    values (p_trip_id, p_pers_id, 'N');
    select max(RESERVATION_ID) into max_id from RESERVATION where PERSON_ID = p_pers_id and trip_id = p_trip_id;
    insert into log(reservation_id, log_date, status)
    values (max_id, current_date, 'N');
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'incorrect data');
end;
```

Do powyższej procedury dodano dodawanie danych do tabeli log.

```
begin
    AddReservation( p_trip_id: 2, p_pers_id: 7);
    commit;
end;
```

	82	2	7	N
LOG_ID	RESERVATION_ID	LOG_DATE	STATUS	
1	82	2023-03-16 15:23:39	N	

Po wykonanie AddReservation, dane poprawnie dodają się do naszych tabel.

## 6.2. Modyfikacja procedury modify\_reservation\_status

```
create or replace procedure modify_reservation_status(p_reservation_id int, p_status char)
as
    tmp char(1);
    av_places int;
begin
    select 1 into tmp from reservation where reservation_id = p_reservation_id;
    if p_status not in ('N', 'P', 'C') then
        raise_application_error(-20002, 'wrong status');
    end if;
    select NO_AVAILABLE_PLACES into av_places from TRIPS T join RESERVATION R on R.TRIP_ID = T.TRIP_ID
        where RESERVATION_ID = p_reservation_id;
    if p_status in ('N', 'P') and av_places = 0 then
        raise_application_error(-20002, 'not enough places');
    end if;
    update reservation
    set status = p_status
    where reservation_id = p_reservation_id;
    insert into log(reservation_id, log_date, status)
        values (p_reservation_id, current_date, p_status);
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'reservation not found');
end;
```

W powyższej procedurze jedyną zmianą do 5.2 jest dodanie nowego rekordu do log.

```
begin
    modify_reservation_status( p_reservation_id: 82, p_status: 'P');
    commit;
end;
```

2	82	2023-03-16 15:29:43	P
---	----	---------------------	---

## 6.3. Wnioski

Tabela log jest ciekawym dodatkiem monitorującym działania w naszej bazie, jednak modyfikacja jej w procedurach może być uciążliwa. Lepszym pomysłem jest wydziedziczenie tego do triggera działającego przy zmianie danych w tabeli Reservations.

## 7. Zmiana podejścia do modyfikacji

### 7.1. Triggery

```

create or replace trigger tr_update_log
  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 tr_dell_log
  before delete
  on RESERVATION
  for each row
begin
  raise_application_error(-20001, 'reservation cant be deleted');
end;

```

tr\_update\_log dodaje rekord do tabeli log, za każdym razem kiedy dodajemy rekord do tabeli, lub zmieniamy jakieś dane.

tr\_dell\_log odpowiada za uniemożliwienie usunięcia rekordu z tabeli reservations.

## 7.2. Procedura AddReservation2

```

create or replace procedure AddReservation2(p_trip_id int, p_pers_id int)
as
  is_person char(1);
  are_trip_ok char(1);
begin
  select 1 into is_person from PERSON where PERSON_ID = p_pers_id;
  select 1 into are_trip_ok from AVAILBLETRIPS
  where TRIP_ID = p_trip_id and TRIP_DATE > current_date;
  insert into RESERVATION(trip_id, person_id, status)
  values (p_trip_id, p_pers_id, 'N');
exception
  when NO_DATA_FOUND then
    raise_application_error(-20001, 'incorrect data');
end;

```

Po wykonaniu polecenia

```

begin
  AddReservation2( p_trip_id: 3, p_pers_id: 2);
  commit;
end;

```

Otrzymujemy w tabeli Reservations

101	3	2	N
-----	---	---	---

Oraz w tabeli log

LOG_ID	RESERVATION_ID	LOG_DATE	STATUS
21	101	2023-03-16 19:51:30	N

### 7.3. Procedura Modify\_reservation\_status

```
create or replace procedure modify_reservation_status2(p_reservation_id int, p_status char)
as
    tmp char(1);
    av_places int;
begin
    select 1 into tmp from reservation where reservation_id = p_reservation_id;
    if p_status not in ('N', 'P', 'C') then
        raise_application_error(-20002, 'wrong status');
    end if;
    select NO_AVAILABLE_PLACES into av_places from TRIPS T join RESERVATION R on R.TRIP_ID = T.TRIP_ID
        where RESERVATION_ID = p_reservation_id;
    if p_status in ('N', 'P') and av_places = 0 then
        raise_application_error(-20002, 'not enough places');
    end if;
    update reservation
    set status = p_status
    where reservation_id = p_reservation_id;
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'reservation not found');
end;
```

```
begin
    modify_reservation_status2( p_reservation_id: 101, p_status: 'P');
    commit;
end;
```

Po wywołaniu powyższej funkcji, modyfikowana jest wartość w tabeli reservations, oraz dodawany jest rekord w tabeli log.

RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
101	3	2	P

LOG_ID	RESERVATION_ID	LOG_DATE	STATUS
22	101	2023-03-16 20:23:33	P
21	101	2023-03-16 19:51:30	N

### 7.4. Wnioski

Takie rozwiązanie odrobinę zmniejsza nam ilość kodu w procedurach oraz pozwala na utrzymanie bardziej czytelnej struktury kodu. Triggery same w sobie nie różnią się znacząco z tymi z MS Sql.

## 8. Kontrola dostępności miejsc w triggerach

### 8.1. Trigger tr\_controll\_places\_add

```
create or replace trigger tr_controll_places_add
before insert
on RESERVATION
for each row
declare
    is_available int;
begin
    select 1 into is_available from AVAILABLETRIPS A where A.TRIP_ID = :new.TRIP_ID;
    if(is_available != 1) then
        raise_application_error(-20001, 'trip is not available');
    end if;
end;
```

Po wprowadzeniu tego triggera, procedura dodająca dane będzie wyglądała następująco:

```
create or replace procedure AddReservation3(p_trip_id int, p_pers_id int)
as
    is_person char(1);
begin
    select 1 into is_person from PERSON where PERSON_ID = p_pers_id;
    insert into RESERVATION(trip_id, person_id, status)
    values (p_trip_id, p_pers_id, 'N');
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'incorrect data');
end;
```

Przy próbie dodania rezerwacji do niedostępnej już wycieczki pojawia nam się oczekiwana odmowa

```
BD_410244> begin
    AddReservation3( 1, 8);
    commit;
end;
[2023-03-21 19:55:44] [72000][20001]
[2023-03-21 19:55:44] ORA-20001: incorrect data
[2023-03-21 19:55:44] ORA-06512: przy "BD_410244.ADDRESERVATION3", linia 10
[2023-03-21 19:55:44] ORA-06512: przy linia 2
[2023-03-21 19:55:44] Position: 0
```

Natomiast dodanie danych do istniejącej tabeli powiedzie się:

```
BD_410244> begin
    AddReservation3( 2, 8);
    commit;
end;
[2023-03-21 19:55:19] completed in 22 ms
```

	RESERVATION_ID ▾ 1	TRIP_ID ▴	PERSON_ID ▴	STATUS ▴
1	123	2	8	N

## 8.2. Tr\_controll\_places\_update

```
create or replace trigger tr_controll_places_update
before update
on RESERVATION
for each row
declare
    is_available int;
begin
    if (:old.status = 'C' and :new.status in ('P', 'N')) then
        select 1 into is_available from AVAILABLETRIPS A where A.TRIP_ID = :new.TRIP_ID;
        if(is_available != 1) then
            raise_application_error(-20001, 'trip is not available');
        end if;
    end if;
end;
```

Po dodaniu tego triggera procedura modyfikująca rezerwacje może wyglądać następująco:

```
create or replace procedure modify_reservation_status3(p_reservation_id int, p_status char)
as
    tmp char(1);
begin
    select 1 into tmp from reservation where reservation_id = p_reservation_id;
    if p_status not in ('N', 'P', 'C') then
        raise_application_error(-20002, 'wrong status');
    end if;
    update reservation
    set status = p_status
    where reservation_id = p_reservation_id;
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'reservation not found');
end;
```

Zmiana statusu wycieczki, której status możemy zmieniać jest dalej bez problemowy, natomiast zmiana statusu na wycieczkę bez wolnych miejsc jest niemożliwy.

```
begin
    modify_reservation_status3( p_reservation_id: 123, p_status: 'P');
    commit;
end
```

LOG_ID ▴	RESERVATION_ID ▾ 1	LOG_DATE ▴	STATUS ▴
41	123	2023-03-21 18:55:21	N
42	123	2023-03-21 19:02:52	P



### 8.3. Wnioski

Dodanie tych triggerów sprawiło że kod procedur stał się czytelniejszy, oraz uzyskaliśmy jaśniejszy podział kodu na część modyfikującą i część sprawdzającą.

## 9. Dodanie pola no\_available\_places

Kolumnę dodaje za pomocą polecenia

```
alter table trip add (no_available_places int default 0)
```

Ustawiając wartość domyślną na 0.

	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	1	Wycieczka do Paryża	Francja	2022-09-12	4	0
2	2	Piękny Kraków	Polska	2023-07-03	10	0
3	3	Znów do Francji	Francja	2023-05-01	15	0
4	4	Heł	Polska	2023-01-01	8	0

Po dodaniu tego pola można zmodyfikować widoki Trips oraz AvailableTrips

```
CREATE OR REPLACE VIEW Trips4 AS
  SELECT T.TRIP_ID, T.country, T.trip_name, T.trip_date, MAX_NO_PLACES no_places, NO_AVAILABLE_PLACES
  from TRIP T
;
```

```
select * from TRIPS4
```

	TRIP_ID	COUNTRY	TRIP_NAME	TRIP_DATE	NO_PLACES	NO_AVAILABLE_PLACES
1	1	Francja	Wycieczka do Paryża	2022-09-12	4	0
2	2	Polska	Piękny Kraków	2023-07-03	10	0
3	3	Francja	Znów do Francji	2023-05-01	15	0
4	4	Polska	Heł	2023-01-01	8	0

```
CREATE OR REPLACE VIEW AvailableTrips4 AS
  SELECT TRIP_ID, country, trip_name, trip_date, no_places, NO_AVAILABLE_PLACES
  FROM Trips4
  where NO_AVAILABLE_PLACES > 0 and current_date < trip_date
;
```

```
select * from AvailableTrips4
```

	TRIP_ID	COUNTRY	TRIP_NAME	TRIP_DATE	NO_PLACES	NO_AVAILABLE_PLACES
--	---------	---------	-----------	-----------	-----------	---------------------

Po stworzeniu procedury liczącej miejsca dostępne na danej wycieczce

```

create or replace procedure count_no_av_places(trip_id int)
as
    tmp int;
    no_places int;
begin
    select 1 into tmp from TRIP T where T.TRIP_ID = count_no_av_places.trip_id;
    if(tmp != 1) then
        raise_application_error(-20002, 'wrong trip id');
    end if;
    select MAX_NO_PLACES - (select count(*) from RESERVATION R where
        R.TRIP_ID = T.TRIP_ID and R.STATUS != 'C')
        into no_places from TRIP T where T.TRIP_ID = count_no_av_places.trip_id;
    update TRIP
    set no_available_places = no_places
    where TRIP_ID = count_no_av_places.trip_id;
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'reservation not found');
end;

```

I wykonaniu jej na wszystkich elementach, np.

```

begin
    count_no_av_places( trip_id: 4);
    commit;
end;

```

	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	1	Wycieczka do Paryża	Francja	2022-09-12	4	3
2	2	Piękny Kraków	Polska	2023-07-03	10	4
3	3	Znów do Francji	Francja	2023-05-01	15	11
4	4	Hel	Polska	2023-01-01	8	7

## 9.1 Modyfikacja funkcji

```

create or replace function FAvailableTrips4(country varchar, date_from date, date_to date)
return available_trips_table
as
    result available_trips_table;
    valid int;
begin
    if FAvailableTrips4.date_to < FAvailableTrips4.date_from then
        raise_application_error(-20001, 'date_to is earlier from date_from');
    end if;
    select count(*) into valid
    from COUNTRIES c
    where c.COUNTRY = FAvailableTrips4.country;
    if valid = 0 then
        raise_application_error(-20001, 'country not found');
    end if;
    select available_trips(t.country, trip_name, trip_date, NO_AVAILABLE_PLACES) bulk collect
    into result
    from TRIP t
    where t.country = FAvailableTrips4.country and NO_AVAILABLE_PLACES > 0
    and trip_date between FAvailableTrips4.date_from and FAvailableTrips4.date_to;
    return result;
end;

```

```

select * from FAvailableTrips4( COUNTRY: 'Polska', DATE_FROM: current_date, DATE_TO: '2024-05-01')

```

	COUNTRY	TRIP_NAME	TRIP_DATE	MAX_NO_PLACES
1	Polska	Piękny Kraków	2023-07-03	4

## 9.2. Modyfikacja Procedur modyfikujących dane

```

create or replace procedure AddReservation4(p_trip_id int, p_pers_id int)
as
    is_person char(1);
begin
    select 1 into is_person from PERSON where PERSON_ID = p_pers_id;
    insert into RESERVATION(trip_id, person_id, status)
    values (p_trip_id, p_pers_id, 'N');
    count_no_av_places(trip_id: p_trip_id);
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'incorrect data');
end;

begin
    AddReservation4( p_trip_id: 3, p_pers_id: 2);
    commit;
end;

```

Po wykonaniu tego kodu, dodana została nowa wartość w tabeli Reservations oraz została na nowo obliczona wartość dodanego pola w tabeli Trip

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	142	3	2	N
2	123	2	8	P
3	101	3	2	P
4	82	2	7	P
5	81	3	9	N
6	61	2	8	C
7	41	2	2	N
8	10	4	9	P
9	9	4	8	C
10	8	3	2	P
11	7	3	8	N
12	6	3	7	C
13	5	2	4	P
14	4	2	4	N
15	3	2	1	P
16	2	1	2	C
17	1	1	1	P

	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	3	Znów do Francji	Francja	2023-05-01	15	10
2	4	Heł	Polska	2023-01-01	8	7
3	2	Piękny Kraków	Polska	2023-07-03	10	4
4	1	Wycieczka do Paryża	Francja	2022-09-12	4	3

Po wykonaniu poniższego kawałka kodu, zmienia się zarówno status rezerwacji, jak i ilość miejsc wolnych na daną wycieczkę.

```

create or replace procedure modify_reservation_status4(p_reservation_id int, p_status char)
as
    tmp char(1);
    p_trip_id int;
begin
    select 1 into tmp from reservation where reservation_id = p_reservation_id;
    if p_status not in ('N', 'P', 'C') then
        raise_application_error(-20002, 'wrong status');
    end if;
    update reservation
    set status = p_status
    where reservation_id = p_reservation_id;
    select TRIP_ID into p_trip_id from RESERVATION where reservation_id = p_reservation_id;
    count_no_av_places( trip_id: p_trip_id);
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'reservation not found');
end;

begin
    modify_reservation_status4( p_reservation_id: 142, p_status: 'C');
    commit;
end;

```

RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
142	3	2	C
123	2	8	P
101	3	2	P
82	2	7	P
81	3	9	N
61	2	8	C
41	2	2	N
10	4	9	P
9	4	8	C

TRIP_ID	TRIP_NAME	COUNTRY	TRIP DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
3	Znów do Francji	Francja	2023-05-01	15	11
4	Hel	Polska	2023-01-01	8	7
2	Piękny Kraków	Polska	2023-07-03	10	4
1	Wycieczka do Paryża	Francja	2022-09-12	4	3

Po lekkiej korekcie, można też zmienić ilość miejsc na dana wycieczkę

```

create or replace procedure modify_no_places4(p_trip_id int, no_places int)
as
    tmp char(1);
    min_places int;
begin
    select 1 into tmp from TRIP where TRIP_ID = p_trip_id;
    select no_places - NO_AVAILABLE_PLACES into min_places from TRIP where TRIP_ID = p_trip_id;
    if min_places > no_places then
        raise_application_error(-20002, 'not enough places');
    end if;
    update TRIP
    set MAX_NO_PLACES = no_places
    where TRIP_ID = p_trip_id;
    count_no_av_places( trip_id: p_trip_id);
end;

begin
    modify_no_places4( p_trip_id: 3, no_places: 5);
    commit;
end;

```

	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	Heł	Polska	2023-01-01	8	7
2	Piękny Kraków	Polska	2023-07-03	10	4
3	Wycieczka do Paryża	Francja	2022-09-12	4	3
4	Znów do Francji	Francja	2023-05-01	5	1

## 10. Zmiana podejścia do pola no\_available\_places

### 10.1. Dodanie triggera na dodanie rezerwacji

```
create or replace trigger tr_add_trip
after insert
on RESERVATION
for each row
declare
    no_places int;
begin
    select NO_AVAILABLE_PLACES into no_places from TRIP t where t.TRIP_ID = :new.TRIP_ID;
    update TRIP
    set no_available_places = no_places - 1
    where TRIP_ID = :new.trip_id;
end;
```

Modyfikuję procedurę AddReservation

```
create or replace procedure AddReservation5(p_trip_id int, p_pers_id int)
as
    is_person char(1);
begin
    select 1 into is_person from PERSON where PERSON_ID = p_pers_id;
    insert into RESERVATION(trip_id, person_id, status)
    values (p_trip_id, p_pers_id, 'N');
exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'incorrect data');
end;

begin
    AddReservation5( p_trip_id: 2, p_pers_id: 9);
    commit;
end;
```

Po jej wykonaniu otrzymuje w tabeli Trip

	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	4	Heł	Polska	2023-01-01	8	7
2	1	Wycieczka do Paryża	Francja	2022-09-12	4	3
3	2	Piękny Kraków	Polska	2023-07-03	10	3
4	3	Znów do Francji	Francja	2023-05-01	5	1

## 10.2. Trigger na modyfikację danych

```
create or replace trigger tr_update_trip
after update
on RESERVATION
for each row
declare
    no_places int;
begin
    select NO_AVAILABLE_PLACES into no_places from TRIP t where t.TRIP_ID = :new.TRIP_ID;
    if (:old.status = 'C' and :new.status in ('P', 'N')) then
        update TRIP
        set no_available_places = no_places - 1
        where TRIP_ID = :new.trip_id;
    end if;
    if (:new.status = 'C' and :old.status in ('P', 'N')) then
        update TRIP
        set no_available_places = no_places + 1
        where TRIP_ID = :new.trip_id;
    end if;
end;
```

```
begin
    modify_reservation_status5( p_reservation_id: 172, p_status: 'C');
    commit;
end;
```

	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	4	Hel	Polska	2023-01-01	8	7
2	2	Piękny Kraków	Polska	2023-07-03	10	4
3	1	Wycieczka do Paryża	Francja	2022-09-12	4	3
4	3	Znów do Francji	Francja	2023-05-01	5	1

## 10.3. Trigger na zmianę liczb miejsc w wycieczce

```
create or replace trigger tr_trip_av_places_update
before update of MAX_NO_PLACES
on Trip
for each row
begin
    if (:new.MAX_NO_PLACES < :old.MAX_NO_PLACES and :old.MAX_NO_PLACES - :new.MAX_NO_PLACES < :old.NO_AVAILABLE_PLACES) then
        raise_application_error(-20001, 'You cant reduce places as much as you try');
    end if;
end;
```

```

create or replace procedure modify_no_places5(p_trip_id int, no_places int)
as
    tmp char(1);
begin
    select 1 into tmp from TRIP where TRIP_ID = p_trip_id;
    update TRIP
    set MAX_NO_PLACES = no_places
    where TRIP_ID = p_trip_id;
    count_no_av_places( trip_id: p_trip_id);
end;

begin
    modify_no_places5( p_trip_id: 3, no_places: 3);
    commit;
end;

```

```

[2023-03-22 13:05:49] [72000][20001]
[2023-03-22 13:05:49] ORA-20001: You cant reduce places as much as you try
[2023-03-22 13:05:49] ORA-06512: przy "BD_410244.TR_TRIP_AV_PLACES_UPDATE", linia 3
[2023-03-22 13:05:49] ORA-04088: błąd w trakcie wykonywania wyzwalacza 'BD_410244.TR_TRIP_AV_PLACES_UPDATE'
[2023-03-22 13:05:49] ORA-06512: przy "BD_410244.MODIFY_NO_PLACES5", linia 6
[2023-03-22 13:05:49] ORA-06512: przy linia 2
[2023-03-22 13:05:49] Position: 0

```

Uzyskano oczekiwany błąd.

	TRIP_ID	TRIP_NAME	COUNTRY	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	1	Wycieczka do Paryża	Francja	2022-09-12	4	3
2	2	Piękny Kraków	Polska	2023-07-03	10	4
3	3	Znów do Francji	Francja	2023-05-01	5	1
4	4	Hel	Polska	2023-01-01	8	7

#### 10.4. Wnioski

Dodanie powyższych triggerów dodaje przejrzystości kodu, oraz zmniejsza objętość procedur.