

Wiktor Woźny, Oracle sprawozdanie

Zadanie 1 – tabele

1) 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
);
```

2) Reservation

```
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;
```

3) Trip

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

alter table trip
    add constraint trip_fk1 foreign key
        (country_id) references country (country_id) enable;
```

4) Log

```
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;
```

5) Country

```
create table country
(
    country_id       int generated always as identity not null,
    country_name     varchar(100),
    constraint country_pk primary key (country_id) enable
);

alter table country
    add constraint country_name_unique unique (country_name);
```

Zadanie 2 – przykładowe dane

Screen zawiera małą część danych wprowadzonych do bazy danych z powodu usunięcia kodu z dodawania niektórych z nich. Widać jednak, że dodawanie rekordów do bazy jest analogiczne.

```
insert into country(country_name)
values ('Francja');

insert into country(country_name)
values ('Niemcy');

insert into country(country_name)
values ('Polska');

insert into trip(trip_name, country_id, trip_date, max_no_places)
values ('Wyciadasdeczka do Paryża', 1, '2022-12-20', 40);

insert into trip(trip_name, country_id, trip_date, max_no_places)
values ('Piękny Kraków', 3, '2023-02-21', 10);

insert into trip(trip_name, country_id, trip_date, max_no_places)
values ('Piękniejszy Rzeszów', 3, '2022-10-27', 5);

insert into trip(trip_name, country_id, trip_date, max_no_places)
values ('Tu nie ma miejsc', 2, '2023-03-11', 0);

insert into person(firstname, lastname)
values ('Marcin', 'Najman');

insert into person(firstname, lastname)
values ('Janek', 'Stawarz');

insert into reservation(trip_id, person_id, status)
values (21, 1, 'C');
```

Zadanie 3 – widoki

- 1) Widok ReservationsDetails (w poleceniu nazwany jako Reservations, nazwa zmieniona, bo taką uważam za bardziej trafną), wyświetla szczegóły wszystkich rezerwacji z tabeli Reservation

```
create view RESERVATIONSDetails as
select C.country_name, T.trip_date, T.trip_name, P.firstname, P.lastname, R.reservation_id, R.status
from trip T
inner join country C on T.country_id = C.country_id
inner join reservation R on T.trip_id = R.trip_id
inner join person P on R.person_id = P.person_id
.
```

	COUNTRY_NAME	TRIP_DATE	TRIP_NAME	FIRSTNAME	LASTNAME	RESERVATION_ID	STATUS
1	Francja	2022-12-20	Wycieczka do Paryża	Marcin	Najman	2	N
2	Polska	2023-02-21	Piękny Kraków	Marcin	Najman	5	P
3	Polska	2022-10-27	Pięknieszy Rzeszów	Marcin	Najman	10	P
4	Polska	2023-02-21	Piękny Kraków	Marcin	Najman	14	C
5	Francja	2023-07-01 12:31:57	res test	Marcin	Najman	61	N
6	Francja	2022-12-20	Wycieczka do Paryża	Wiktor	Woźny	11	C
7	Niemcy	2023-03-30	Ta będzie potem	Adam	Misztal	1	N
8	Francja	2022-12-20	Wycieczka do Paryża	Adam	Misztal	21	C
9	Niemcy	2023-04-22 17:32:57	Tu było jedno miejsce	Adam	Misztal	41	N
10	Niemcy	2023-03-30	Ta będzie potem	Remek	Kozicki	12	N
11	Polska	2022-10-27	Pięknieszy Rzeszów	Witek	Strzeboński	9	N
12	Niemcy	2022-09-30	Browar w Monachium	Krzychu	Usnarski	8	C
13	Niemcy	2023-03-30	Ta będzie potem	Krzychu	Usnarski	13	P
14	Polska	2023-02-21	Piękny Kraków	Robert	Lewandowski	6	P
15	Francja	2023-07-01 12:31:57	res test	Robert	Lewandowski	81	C
16	Niemcy	2022-09-30	Browar w Monachium	Adrian	Prędkie	7	C
17	Francja	2023-07-01 12:31:57	res test	Adrian	Prędkie	62	N

- 2) Pomocniczy widok AvailablePlaces, dla każdej wycieczki zlicza ilość dostępnych jeszcze miejsc.

```
create or replace view AvailablePlaces (trip_id, av_places) as
select T.trip_id,
       av_places: T.max_no_places - (select count(R.reservation_id)
                                     from reservation R
                                     where R.trip_id = T.trip_id
                                     and R.status != 'C')
from trip T;
```

	TRIP_ID	AV_PLACES
1	41	120
2	61	0
3	121	8
4	1	14
5	21	8
6	22	3
7	23	11
8	81	0
9	141	10
10	101	40

- 3) Widok TripsDetails (w poleceniu nazwany jako Trips, nazwa zmieniona, bo taką uważam za bardziej trafną), wyświetla szczegóły wszystkich wycieczek z tabeli Trip, korzysta z pomocniczego widoku AvailablePlaces.

```
create or replace view TripsDetails (country, trip_date, trip_name, no_places, no_available_places) as
select C.country_name, T.trip_date, T.trip_name, T.max_no_places, A.av_places
from trip T
    inner join AVAILABLEPLACES A on A.trip_id = T.trip_id
    inner join country C on C.country_id = T.country_id;
```

	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NO_AVAILABLE_PLACES
1	Francja	2023-07-01 12:31:57	res test	10	8
2	Francja	2022-12-20	Wycieczka do Paryża	15	14
3	Francja	2022-12-20	Wyciasdasdeczka do Paryża	40	40
4	Niemcy	2023-07-03 10:58:04	Ona będzie później	10	10
5	Niemcy	2023-03-11	Tu nie ma miejsc	0	0
6	Niemcy	2023-03-30	Ta będzie potem	123	120
7	Niemcy	2022-09-30	Browar w Monachium	11	11
8	Niemcy	2023-04-22 17:32:57	Tu było jedno miejsce	1	0
9	Polska	2022-10-27	Pięknniejszy Rzeszów	5	3
10	Polska	2023-02-21	Piękny Kraków	10	8

- 4) Widok AvailableTrips, wyświetla wycieczki (takie same dane jak w przypadku widoku TripsDetails), które są dostępne (czyli takich, których liczba dostępnych miejsc jest większa od 0 i jeszcze się nie rozpoczęły), widok korzysta z widoku TripsDetails w celu uniknięcia powtarzania kodu.

```
create or replace view AvailableTrips (country, trip_date, trip_name, no_places, no_available_places) as
select *
from TripsDetails TD
where TD.no_available_places > 0
    and TD.trip_date > current_date;
```

	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NO_AVAILABLE_PLACES
1	Francja	2023-07-01 12:31:57	res test	10	8
2	Niemcy	2023-07-03 10:58:04	Ona będzie później	10	10

- 5) Pomocniczy widok AvailableTripsNoDateCheck, wyświetla wszystkie wycieczki, dla których liczba dostępnych jeszcze miejsc jest większa od 0 (widok używany potem w jednej z funkcji).

```
create or replace view AvailableTripsNoDateCheck (country, trip_date, trip_name, no_places, no_available_places) as
select *
from TripsDetails TD
where TD.no_available_places > 0;
```

	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NO_AVAILABLE_PLACES
1	Francja	2023-07-01 12:31:57	res test	10	8
2	Francja	2022-12-20	Wycieczka do Paryża	15	14
3	Francja	2022-12-20	Wyciasdasdeczka do Paryża	40	40
4	Niemcy	2023-07-03 10:58:04	Ona będzie później	10	10
5	Niemcy	2023-03-30	Ta będzie potem	123	120
6	Niemcy	2022-09-30	Browar w Monachium	11	11
7	Polska	2022-10-27	Pięknniejszy Rzeszów	5	3
8	Polska	2023-02-21	Piękny Kraków	10	8

Zadanie 4 – funkcje/procedury

1) Obiekty potrzebne funkcjom, procedurom + Tabele

a) trip_participant (trip_participant_table)

```
create or replace type trip_participant as object
(
    firstname      varchar2(50),
    lastname       varchar2(50),
    reservation_id number,
    status         char(1)
);
```

```
create or replace type trip_participants_table is table of trip_participant;
```

b) person_reservation (person_reservation_table)

```
create or replace type person_reservation as object
(
    country        varchar2(100),
    trip_date      date,
    trip_name      varchar2(100),
    reservation_id number,
    status         char(1)
);
```

```
create or replace type person_reservations_table as table of person_reservation;
```

c) available_trips (available_trips_table)

```
create or replace type available_trips as object
(
    trip_date      date,
    trip_name      varchar2(100),
    no_places      number,
    no_available_places number
);
```

```
create or replace type available_trips_table as table of available_trips;
```

2) Procedury sprawdzające poprawność danych

- a) trip_exist, sprawdza czy wycieczka o podanym id istnieje w tabeli Trip, jeżeli nie, wznoszony jest błąd

```
create or replace procedure trip_exist(trip_id number)
as
    tmp char(1);
begin

    select 1 into tmp from trip t where T.trip_id = trip_exist.trip_id;

exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'trip not found');

end;
```

- b) person_exist, sprawdza czy osoba o podanym id istnieje w tabeli Person, jeżeli nie, wznoszony jest błąd

```
create or replace procedure person_exist(person_id number)
as
    tmp char(1);
begin

    select 1 into tmp from person P where P.person_id = person_exist.person_id;

exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'person not found');

end;
```

- c) country_exist, sprawdza czy kraj o podanej nazwie istnieje w tabeli Country, jeżeli nie, wznoszony jest błąd

```
create or replace procedure country_exist(country_name varchar2)
as
    tmp char(1);
begin

    select 1 into tmp from country C where C.country_name = country_exist.country_name;

exception
    when NO_DATA_FOUND then
        raise_application_error(-20001, 'there is no such country in database');

end;
```

- d) `is_trip_available`, sprawdza czy wycieczka o podanym id jest jeszcze dostępna (tj. są jeszcze dostępne miejsca i jeszcze się nie zaczęła), jeżeli nie, wznoszony jest błąd

```
create or replace procedure is_trip_available(trip_id number)
as
    tmp char(1);
begin

    select 1
    into tmp
    from AvailableTrips A
    |       inner join trip t on A.trip_name = t.trip_name
    where t.trip_id = is_trip_available.trip_id;

exception
    when NO_DATA_FOUND then
    |     raise_application_error(-20001, 'trip unavailable');

end;
```

- e) `reservation_exist`, sprawdza czy rezerwacja o podanym id istnieje w tabeli `Reservation`, jeżeli nie, wznoszony jest błąd

```
create or replace procedure reservation_exist(reservation_id number)
as
    tmp char(1);
begin

    select 1 into tmp from reservation R where R.reservation_id = reservation_exist.reservation_id;

exception
    when NO_DATA_FOUND then
    |     raise_application_error(-20001, 'reservation not found');

end;
```


3) Funkcje ułatwiające dostęp do danych

- a) TripParticipants, funkcja zwraca tabelę zawierającą dane uczestników wycieczki o podanym id

```
create or replace function TripParticipants(trip_id number)
return trip_participants_table
as
result trip_participants_table;
begin

--      throws an error
trip_exist( trip_id: TripParticipants.trip_id);

select trip_participant(P.firstname, P.lastname, R.reservation_id, R.status) bulk collect
into result
from person P
    inner join reservation R on R.person_id = P.person_id
where R.trip_id = TripParticipants.trip_id;

return result;
end;
```

```
BD_410434> select *
from TripParticipants(121)
```

	FIRSTNAME	LASTNAME	RESERVATION_ID	STATUS
1	Marcin	Najman	61	N
2	Robert	Lewandowski	81	C
3	Adrian	Prędkie	62	N

- b) PersonReservations, funkcja zwraca tabelę zawierającą informacje o rezerwacjach przypisanych do osoby o podanym id

```
create or replace function PersonReservations(person_id number)
return person_reservations_table
as
result person_reservations_table;
begin

person_exist( person_id: PersonReservations.person_id);

select person_reservation(C.country_name, T.trip_date, T.trip_name, R.reservation_id, R.status) bulk collect
into result
from trip T
    inner join country C on C.country_id = T.country_id
    inner join reservation R on R.trip_id = T.trip_id
    inner join person P on P.person_id = R.person_id
where P.person_id = PersonReservations.person_id;

return result;
end;
```

```
BD_410434> select *
from PersonReservations(1)
```

	COUNTRY	TRIP_DATE	TRIP_NAME	RESERVATION_ID	STATUS
1	Francja	2022-12-20	Wycieczka do Paryża	2	N
2	Polska	2023-02-21	Piękny Kraków	5	P
3	Polska	2022-10-27	Pięknieszy Rzeszów	10	P
4	Polska	2023-02-21	Piękny Kraków	14	C
5	Francja	2023-07-01 12:31:57	res test	61	N

- c) AvailableTripsFunc (w poleceniu nazwane jako AvailableTrips, nazwa zmieniona ze względu na konflikt nazw z widokiem powyżej), funkcja zwraca tabelę zawierającą dane dostępnych wycieczek z kraju podanego w argumencie funkcji, dodatkowo jako argumenty przyjmuje zakres dat, w których wycieczki te mają się odbywać.

```
create or replace function AvailableTripsFunc(country varchar2, date_from date, date_to date)
    return available_trips_table
as
    result available_trips_table;
begin

    if AvailableTripsFunc.date_from > AvailableTripsFunc.date_to then
        raise_application_error(-20001, 'wrong date range');
    end if;

    country_exist( country_name: AvailableTripsFunc.country);

    select available_trips(V.trip_date, V.trip_name, V.no_places, V.no_available_places) bulk collect
    into result
    from AvailableTripsNoDateCheck V
    where V.trip_date >= AvailableTripsFunc.date_from
        and V.trip_date <= AvailableTripsFunc.date_to
        and AvailableTripsFunc.country = V.country;

    return result;

end;
```

```
BD_410434> select *
            from AvailableTripsFunc('Polska', '2022-05-12', '2023-03-10')
```

	TRIP_DATE	TRIP_NAME	NO_PLACES	NO_AVAILABLE_PLACES
1	2023-02-21	Piękny Kraków	10	8
2	2022-10-27	Pięknieszy Rzeszów	5	3

Zadanie 5 –procedury modyfikujące dane

- 1) AddReservation, procedura dodaje do tabeli Reservation nową rezerwację, jako argumenty przyjmuje id wycieczki oraz id osoby, automatycznie dodaje nowy rekord do tabeli Log.

```
create or replace procedure AddReservation(trip_id number, person_id number)
as
    new_res_id number;
begin
    person_exist( person_id: AddReservation.person_id);
    trip_exist( trip_id: AddReservation.trip_id);
    is_trip_available( trip_id: AddReservation.trip_id);

    insert into reservation
    values (default, AddReservation.trip_id, AddReservation.person_id, 'N')
    returning reservation_id into new_res_id;

    insert into log values (default, new_res_id, current_date, 'N');

end;
```

```
BD_410434> begin
    AddReservation(121, 8);
    commit;
end;
```

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	61	121	1	N
2	81	121	8	C
3	101	121	8	N
4	62	121	9	N
5	41	81	3	N

Widać nową rezerwację dodaną dla osoby z id 8 o statusie N.

- 2) ModifyReservationStatus, procedura zmienia status rezerwacji o podanym id na status podany w argumencie procedury

```
create or replace procedure ModifyReservationStatus(reservation_id number, status char)
as
    current_status char;
    trip_id        number;
begin
    reservation_exist( reservation_id: ModifyReservationStatus.reservation_id);

    select R.status
    into current_status
    from reservation R
    where R.reservation_id = ModifyReservationStatus.reservation_id;

    select R.trip_id
    into ModifyReservationStatus.trip_id
    from reservation R
    where R.reservation_id = ModifyReservationStatus.reservation_id;

    if ModifyReservationStatus.status not in ('N', 'P', 'C') then
        raise_application_error(-20001, 'wrong reservation status');
    end if;

    -- if current status is 'C' and we want to change it, we have to check if there are any free places

    if current_status = 'C' and ModifyReservationStatus.status != 'C' then
        is_trip_available( trip_id: ModifyReservationStatus.trip_id);
    end if;

    insert into log values (default, ModifyReservationStatus.reservation_id, current_date, current_status);

    update reservation R
    set R.status = ModifyReservationStatus.status
    where R.reservation_id = ModifyReservationStatus.reservation_id;
end;
```

```
BD_410434> begin
    ModifyReservationStatus(101, 'C');
    commit;
end;
```

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	61	121	1	N
2	81	121	8	C
3	101	121	8	C
4	62	121	9	N

jak widać rezerwacja 101 zmieniła status

3) ModifyNoPlaces, procedura modyfikuje maksymalną ilość miejsc wycieczki dla podanego id wycieczki

```

create or replace procedure ModifyNoPlaces(trip_id number, no_places number)
as
    currplaces number;
begin
    trip_exist( trip_id: ModifyNoPlaces.trip_id);

    select count(*)
    into currplaces
    from trip T
        inner join reservation R on R.trip_id = T.trip_id
    where T.trip_id = ModifyNoPlaces.trip_id
        and R.status != 'C';

    if ModifyNoPlaces.no_places < currplaces then
        raise_application_error(-20001, 'you can not reduce number of places');
    end if;

    update trip T
    set T.max_no_places = ModifyNoPlaces.no_places
    where T.trip_id = ModifyNoPlaces.trip_id;
end;

BD_410434> begin
    ModifyNoPlaces(1, 16);
end;

```

	TRIP_ID	TRIP_NAME	COUNTRY_ID	TRIP_DATE	MAX_NO_PLACES	NO_AVAILABLE_PLACES
1	41	Ta będzie potem	2	2023-03-30	123	120
2	61	Tu nie ma miejsc	2	2023-03-11	0	0
3	121	res test	1	2023-07-01 12:31:57	10	6
4	1	Wycieczka do Paryża	1	2022-12-20	16	14

Max_no_places zostało zmienione na 16

Zadanie 6 – dodanie tabeli log

Kod, który dodał tabelę Log do bazy danych znajduje się w zadaniu 1, tak samo procedury z zadania 5 zawierają już inserty, które odpowiednio modyfikują tabelę Log.

	LOG_ID	RESERVATION_ID	LOG_DATE	1	STATUS
1	86	101	2023-04-04 12:50:28		N
2	87	101	2023-04-04 12:50:28		C
3	81	101	2023-04-04 12:36:38		N
4	84	101	2023-04-04 12:36:38		N
5	83	101	2023-04-04 12:36:38		N
6	82	101	2023-04-04 12:36:38		N

Widać tu wszystkie modyfikacje pokazywane wcześniej w sprawozdaniu.

Zadanie 7 – triggery obsługujące tabelę Log

- 1) add_or_modify_reservation_log_trigger, trigger odpowiednio modyfikuje tabelę Log przy dodawaniu lub modyfikowaniu już istniejącego rekordu w tabeli Reservation

```
create or replace trigger add_or_modify_reservation_log_trigger
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;
```

Poniżej można zobaczyć zmodyfikowane procedury po wprowadzeniu triggera. Widać, że uniknęliśmy pisania kodu o modyfikacji tabeli Log w procedurze, dzięki czemu zastosowaliśmy się do zasady pojedynczej odpowiedzialności.

```
create or replace procedure AddReservation2(trip_id number, person_id number)
as
begin
person_exist( person_id: AddReservation2.person_id);
trip_exist( trip_id: AddReservation2.trip_id);
is_trip_available( trip_id: AddReservation2.trip_id);

insert into reservation
values (default, AddReservation2.trip_id, AddReservation2.person_id, 'N');

end;

create or replace procedure ModifyReservationStatus2(reservation_id number, status char)
as
current_status char;
trip_id number;
begin
reservation_exist( reservation_id: ModifyReservationStatus2.reservation_id);

select R.status
into current_status
from reservation R
where R.reservation_id = ModifyReservationStatus2.reservation_id;

select R.trip_id
into ModifyReservationStatus2.trip_id
from reservation R
where R.reservation_id = ModifyReservationStatus2.reservation_id;

if ModifyReservationStatus2.status not in ('N', 'P', 'C') then
raise_application_error(-20001, 'wrong reservation status');
end if;

-- if current status is 'C' and we want to change it, we have to check if there are any free places

if current_status = 'C' and ModifyReservationStatus2.status != 'C' then
is_trip_available( trip_id: ModifyReservationStatus2.trip_id);
end if;

update reservation R
set R.status = ModifyReservationStatus2.status
where R.reservation_id = ModifyReservationStatus2.reservation_id;

end;
```

```
BD_410434> begin
    AddReservation2(121, 4);
    commit;
end;
```

	LOG_ID	RESERVATION_ID	LOG_DATE	1	STATUS
1	104	122	2023-04-04 13:14:12		N

Jak widać, trigger po dodaniu rezerwacji działa poprawnie.

```
BD_410434> begin
    ModifyReservationStatus2(122, 'C');
end;
```

	LOG_ID	RESERVATION_ID	LOG_DATE	1	STATUS
1	107	122	2023-04-04 13:15:47		C

Tak samo po zmianie statusu rezerwacji.

2) delete_reservation_trigger, trigger zabrania usunięcia rezerwacji

```
create or replace trigger delete_reservation_trigger
before delete
on reservation
for each row
begin
    raise_application_error(-20001, 'it is forbidden to remove reservation');
end;
```

```
BD_410434> begin
    delete from reservation where reservation_id = 122;
    commit;
end;
```

```
[72000][20001]
ORA-20001: it is forbidden to remove reservation
ORA-06512: przy "BD_410434.DELETE_RESERVATION_TRIGGER", linia 2
ORA-04088: błąd w trakcie wykonywania wyzwalacza 'BD_410434.DELETE_RESERVATION_TRIGGER'
ORA-06512: przy lini ...
```

Trigger działa poprawnie przy próbie usunięcia rezerwacji.

Zadanie 9 – dodanie oraz obsługa redundantnego pola no_available_places

1) dodanie pola do tabeli

```
alter table trip
  add no_available_places int;
```

2) procedura uaktualniająca tabelę

```
create or replace procedure update_trip
as
begin
  update trip T
  set T.no_available_places = T.max_no_places - (select trip_id: count(*)
                                                from reservation R
                                                where R.trip_id = T.trip_id
                                                and R.status != 'C');
end;

begin
  update_trip();
  commit;
end;
```

3) aktualizacja widoków i funkcji (wraz z dowodem poprawności działania)

```
create or replace view TripsDetails4 (country, trip_date, trip_name, no_places, no_available_places) as
select C.country_name, T.trip_date, T.trip_name, T.max_no_places, T.no_available_places
from trip T
inner join country C on C.country_id = T.country_id;
```

	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NO_AVAILABLE_PLACES
1	Francja	2023-07-01 12:31:57	res test	10	4
2	Francja	2022-12-20	Wycieczka do Paryża	16	14
3	Francja	2022-12-20	Wyciasdasdeczka do Paryża	40	40
4	Niemcy	2023-07-03 10:58:04	Ona będzie później	10	10
5	Niemcy	2023-03-11	Tu nie ma miejsc	0	0
6	Niemcy	2023-03-30	Ta będzie potem	123	120
7	Niemcy	2022-09-30	Browar w Monachium	11	11
8	Niemcy	2023-04-22 17:32:57	Tu było jedno miejsce	1	0
9	Polska	2022-10-27	Pięknniejszy Rzeszów	5	3
10	Polska	2023-02-21	Piękny Kraków	10	8

```

create or replace view AvailableTrips4 (country, trip_date, trip_name, no_places, no_available_places) as
select *
from TripsDetails4 TD
where TD.no_available_places > 0
and TD.trip_date > current_date;

```

	COUNTRY	TRIP_DATE	TRIP_NAME	NO_PLACES	NO_AVAILABLE_PLACES
1	Francja	2023-07-01 12:31:57	res test	10	7
2	Niemcy	2023-07-03 10:58:04	Ona będzie później	10	10

```

create or replace function AvailableTripsFunc4(country varchar2, date_from date, date_to date)
return available_trips_table
as
result available_trips_table;
begin

if AvailableTripsFunc4.date_from > AvailableTripsFunc4.date_to then
raise_application_error(-20001, 'wrong date range');
end if;

country_exist( country_name: AvailableTripsFunc4.country);

select available_trips(T.trip_date, T.trip_name, T.max_no_places, T.no_available_places) bulk collect
into result
from trip T
inner join country C on T.country_id = C.country_id
where T.trip_date >= AvailableTripsFunc4.date_from
and T.trip_date <= AvailableTripsFunc4.date_to
and AvailableTripsFunc4.country = C.country_name
and T.no_available_places > 0;

return result;

end;

```

	TRIP_DATE	TRIP_NAME	NO_PLACES	NO_AVAILABLE_PLACES
1	2023-02-21	Piękny Kraków	10	8
2	2022-10-27	Piękniejszy Rzeszów	5	3

Dzięki redundantnemu polu nie było konieczności używania pomocniczych widoków, jednak czy było warto dla zaśmiecania bazy dodatkowym polem? Według mnie nie.

4) aktualizacja procedur (wraz z dowodem poprawności działania)

```
create or replace procedure is_trip_available4(trip_id number)
as
    tmp char(1);
begin
    select 1
    into tmp
    from trip T
    where t.trip_id = is_trip_available4.trip_id
        and t.no_available_places > 0
        and t.trip_date > current_date;

    exception
        when NO_DATA_FOUND then
            raise_application_error(-20001, 'trip unavailable');

end;
```

```
BD_410434> begin
            is_trip_available4(123);
        end;
```

```
[72000][20001]
ORA-20001: trip unavailable
ORA-06512: przy "BD_410434.IS_TRIP_AVAILABLE4", linia 13
ORA-06512: przy linia 2
Position: 0
```

```
create or replace procedure AddReservation4(trip_id number, person_id number)
as
begin
    person_exist( person_id: AddReservation4.person_id);
    trip_exist( trip_id: AddReservation4.trip_id);
    is_trip_available4( trip_id: AddReservation4.trip_id);

    insert into reservation
    values (default, AddReservation4.trip_id, AddReservation4.person_id, 'N');

    update trip T
    set T.no_available_places = T.no_available_places - 1
    where AddReservation4.trip_id = T.trip_id;

end;
```

```
BD_410434> begin
            AddReservation4(121, 4);
            commit;
        end;
```

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	61	121	1	N
2	81	121	8	C
3	141	121	4	N
...

```

create or replace procedure ModifyReservationStatus4(reservation_id number, status char)
as
    current_status char;
    trip_id        number;
begin
    reservation_exist( reservation_id: ModifyReservationStatus4.reservation_id);

    select R.status
    into current_status
    from reservation R
    where R.reservation_id = ModifyReservationStatus4.reservation_id;

    select R.trip_id
    into ModifyReservationStatus4.trip_id
    from reservation R
    where R.reservation_id = ModifyReservationStatus4.reservation_id;

    if ModifyReservationStatus4.status not in ('N', 'P', 'C') then
        raise_application_error(-20001, 'wrong reservation status');
    end if;

    -- if current status is 'C' and we want to change it, we have to check if there are any free places

    if current_status = 'C' and ModifyReservationStatus4.status != 'C' then
        is_trip_available4( trip_id: ModifyReservationStatus4.trip_id);

        update trip T
        set T.no_available_places = T.no_available_places - 1
        where ModifyReservationStatus4.trip_id = T.trip_id;
    end if;

    if current_status != 'C' and ModifyReservationStatus4.status = 'C' then
        update trip T
        set T.no_available_places = T.no_available_places + 1
        where ModifyReservationStatus4.trip_id = T.trip_id;
    end if;

    update reservation R
    set R.status = ModifyReservationStatus4.status
    where R.reservation_id = ModifyReservationStatus4.reservation_id;
end;

```

```

BD_410434> begin
            ModifyReservationStatus4(141, 'C');
            commit;
        end;

```

	RESERVATION_ID	TRIP_ID	PERSON_ID	STATUS
1	61	121	1	N
2	81	121	8	C
3	141	121	4	C

Zadanie 10 – dodanie triggerów oraz modyfikacja procedur po dodaniu redundantnego pola no_available_places

- 1) add_reservation_log_trigger2, trigger dodaje rekord do tabeli Log i uaktualnia pole no_available_places w tabeli Trip (wszystko oczywiście po dodaniu nowej rezerwacji do tabeli Reservation)

```
create or replace trigger add_reservation_log_trigger2
after insert
on reservation
for each row
begin
insert into log (reservation_id, log_date, status)
values (:new.reservation_id, current_date, :new.status);

update trip T
set T.no_available_places = T.no_available_places - 1
where T.trip_id = :new.trip_id;
end;
```

- 2) modify_reservation_log_trigger2, trigger dodaje rekord do tabeli Log i uaktualnia pole no_available_places w tabeli Trip (wszystko oczywiście po modyfikacji statusu rezerwacji)

```
create or replace trigger modify_reservation_log_trigger2
after update
on reservation
for each row
declare
to_add number;
begin
insert into log (reservation_id, log_date, status)
values (:new.reservation_id, current_date, :new.status);

if :new.status = 'C' then
to_add = 1;
else
to_add = 0;
end if;

update trip T
set T.no_available_places = T.no_available_places + to_add
where T.trip_id = :new.trip_id;
end;
```

- 3) change_no_of_places_trigger, trigger uaktualnia pole no_of_available_places po modyfikacji pola max_no_places w tabeli Trip

```
create or replace trigger change_no_of_places_trigger
after update of max_no_places
on trip
for each row
begin
update trip T
set T.no_available_places = T.no_available_places + (:new.max_no_places - T.max_no_places);
end;
```

- 4) zmiany procedur modyfikujących dane:

```
create or replace procedure AddReservation5(trip_id number, person_id number)
as
begin
person_exist( person_id: AddReservation5.person_id);
trip_exist( trip_id: AddReservation5.trip_id);
is_trip_available4( trip_id: AddReservation5.trip_id);

insert into reservation
values (default, AddReservation5.trip_id, AddReservation5.person_id, 'N');
end;

create or replace procedure ModifyReservationStatus5(reservation_id number, status char)
as
current_status char;
trip_id number;
begin
reservation_exist( reservation_id: ModifyReservationStatus5.reservation_id);

select R.status
into current_status
from reservation R
where R.reservation_id = ModifyReservationStatus5.reservation_id;

select R.trip_id
into ModifyReservationStatus5.trip_id
from reservation R
where R.reservation_id = ModifyReservationStatus5.reservation_id;

if ModifyReservationStatus5.status not in ('N', 'P', 'C') then
raise_application_error(-20001, 'wrong reservation status');
end if;

-- if current status is 'C' and we want to change it, we have to check if there are any free places

if current_status = 'C' and ModifyReservationStatus5.status != 'C' then
is_trip_available4( trip_id: ModifyReservationStatus5.trip_id);

update trip T
set T.no_available_places = T.no_available_places - 1
where ModifyReservationStatus5.trip_id = T.trip_id;
end if;

update reservation R
set R.status = ModifyReservationStatus5.status
where R.reservation_id = ModifyReservationStatus5.reservation_id;
end;
```

```
create or replace procedure ModifyNoPlaces5(trip_id number, no_places number)
as
    currplaces number;
begin
    trip_exist( trip_id: ModifyNoPlaces5.trip_id);

    select T.max_no_places - T.no_available_places
    into currplaces
    from trip T
    where T.trip_id = ModifyNoPlaces5.trip_id;

    if ModifyNoPlaces5.no_places < currplaces then
        raise_application_error(-20001, 'you can not reduce number of places');
    end if;

    update trip T
    set T.max_no_places = ModifyNoPlaces5.no_places
    where T.trip_id = ModifyNoPlaces5.trip_id;
end;
```