Tabele

LOGS

```
create table LOGS
(
   ID             NUMBER generated as identity
       constraint LOG_PK
           primary key,
   RESERVATION_ID NUMBER,
   LOG_DATE       DATE,
   STATUS         CHAR,
   NO_PLACES      NUMBER
)
```

Widoki

AVAILABLE_TRIPS

```
create view AVAILABLE_TRIPS as
select
   COUNTRY,
   TRIP_DATE,
   NAME,
   MAX_NO_PLACES,
   NO_AVAILABLE_PLACES
from TRIPS
where NO_AVAILABLE_PLACES>0 and TRIP_DATE>CURRENT_DATE
```

RESERVATIONS

```
create or replace view RESERVATIONS as
select
   country,
   trip_date,
   T.NAME as trip_name,
   firstname,
   lastname,
   reservation_id,
   T.trip_id,
   P.PERSON_ID,
```

```
   no_places,status
from
   RESERVATION
   inner join PERSON P on RESERVATION.PERSON_ID = P.PERSON_ID
   inner join TRIP T on T.TRIP_ID = RESERVATION.TRIP_ID
```

TRIPS

```
create view TRIPS as
select
   country,
   trip_date,
   R.TRIP_ID,
   T.name,
   max_no_places,
   MAX_NO_PLACES-NO_PLACES as no_available_places
from
   RESERVATION R inner join TRIP T on T.TRIP_ID = R.TRIP_ID
```

Typy

available_trip

```
create type available_trip as object
(
   country varchar2(50),
   trip_date date,
   trip_name varchar2(100),
   max_no_places int,
   no_available_places int
)
```

person_reservation

```
create type person_reservation as object
(
   reservation_id int,
   person_id int,
   firstname varchar2(50),
   lastname varchar2(50),
   trip_name varchar2(100),
   country Varchar2(50),
```

```
   trip_date date,
   no_places int,
   status char(1)
)
```

trip_participant

```
create type trip_participant as object
(
   reservation_id int,
   trip_id int,
   firstname varchar2(50),
   lastname varchar2(50),
   trip_name varchar2(100),
   country Varchar2(50),
   trip_date date,
   no_places int,
   status char(1)
)
```

Kolekcje

```
create type AVAILABLE_TRIP_TABLE as table of AVAILABLE_TRIP
```

```
create type PERSON_RESERVATION_TABLE as table of PERSON_RESERVATION
```

```
create type TRIP_PARTICIPANT_TABLE as table of TRIP_PARTICIPANT
```

Funkcje

available_trips_to

```
create or replace function available_trips_to(country varchar2,
date_from date, date_to date)
   return available_trip_table
as
   result available_trip_table;
begin
   select available_trip(
       AVAILABLE_TRIPS.COUNTRY,
```

```
        TRIP_DATE,
        NAME,
        MAX_NO_PLACES,
        NO_AVAILABLE_PLACES
               )
    bulk collect into result
    from AVAILABLE_TRIPS
    where AVAILABLE_TRIPS.COUNTRY = available_trips_to.country
    and AVAILABLE_TRIPS.TRIP_DATE between date_from and date_to;

    return result;
end;
```

trip_participants

```
create function trip_participants(trip_id int)
    return trip_participant_table
as
    result trip_participant_table;
begin
    select trip_participant(
        RESERVATION_ID,
        R.TRIP_ID,
        FIRSTNAME,
        LASTNAME,
        TRIP_NAME,
        COUNTRY,
        TRIP_DATE,
        NO_PLACES,
        STATUS
        )
    bulk collect into result
    from RESERVATIONS R
    where R.TRIP_ID = trip_participants.trip_id;

    return result;
end;
```

person_reservations

```
create function person_reservations(person_id int)
    return person_reservation_table
as
    result person_reservation_table;
```

```
   valid int;
begin
   select count(*) into valid from PERSON where PERSON.PERSON_ID =
person_reservations.PERSON_ID;

   if valid = 0
   then
       raise_application_error(-20001, 'person not found');
   end if;

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

   return result;
end;
```

Procedury

add_reservation

```
create procedure add_reservation(trip_id int, person_id int, no_places
int)
as
   value int;
begin
   select count(*) into value from TRIP where
add_reservation.trip_id=TRIP.TRIP_ID;
   if value = 0 then
       raise_application_error(-20001, 'trip not found');
   end if;
   select count(*) into value from PERSON where
add_reservation.person_id=PERSON.PERSON_ID;
```

```
    if value = 0 then
        raise_application_error(-20001, 'person not found');
    end if;
    select count(*) into value from PERSON where
add_reservation.person_id=PERSON.PERSON_ID;
    if value = 0 then
        raise_application_error(-20001, 'person not found');
    end if;

    insert into RESERVATION
        (TRIP_ID, PERSON_ID, STATUS, NO_PLACES)
    values
        (trip_id, person_id, 'n', no_places);
end;
```

modify_max_no_places

```
create procedure modify_max_no_places(trip_id int, no_places int)
as
    value int;
begin
    select count(*) into value from TRIP where
modify_max_no_places.trip_id=TRIP.TRIP_ID;
    if value = 0 then
        raise_application_error(-20001, 'trip not found');
    end if;

    update TRIP
        set MAX_NO_PLACES = no_places
    where TRIP_ID = modify_max_no_places.trip_id;
end;
```

modify_reservation

```
create procedure modify_reservation(reservation_id int, no_places int)
as
    value int;
begin
    select count(*) into value from RESERVATION where
modify_reservation.reservation_id=RESERVATION.RESERVATION_ID;
    if value = 0 then
        raise_application_error(-20001, 'reservation not found');
    end if;
```

```
    update RESERVATION
        set NO_PLACES = no_places
    where RESERVATION_ID = modify_reservation.reservation_id;
end;
```

modify_reservation_status

```
create procedure  modify_reservation_status(p_reservation_id int,
p_status char)
as
    old_status char(1);
    places varchar2(50);
begin
    select STATUS into old_status from RESERVATION where RESERVATION_ID =
p_reservation_id;
    select NO_AVAILABLE_PLACES into places from AVAILABLE_TRIPS;
    if old_status = 'c' and p_status = 'n' then
        if places = 0 then
            raise_application_error(-20001, 'no available places for the
trip');
        end if;
    end if;

    update RESERVATION
        set STATUS = p_status
    where RESERVATION_ID = p_reservation_id;
end;
```

Triggery

TR_INSERT_RESERVATION_PLACES

```
create trigger TR_INSERT_RESERVATION_PLACES
    before insert
    on RESERVATION
    for each row
declare
    places int;
begin
    select NO_AVAILABLE_PLACES into places from TRIPS where TRIPS.TRIP_ID
= :new.TRIP_ID;

    if :new.NO_PLACES < places then
```

```
        raise_application_error(-20001, 'trip doest not have enough
available places');
    end if;
end;
```

TR_UPDATE_LOG

```
create trigger TR_UPDATE_LOG
    after insert or update
    on RESERVATION
    for each row
begin
    insert into LOGS (RESERVATION_ID, LOG_DATE, STATUS, NO_PLACES)
    values (:new.RESERVATION_ID, current_date, :new.STATUS,
:new.NO_PLACES);
end;
```

TR_UPDATE_RESERVATION_STATUS

```
create trigger TR_UPDATE_RESERVATION_STATUS
    before update
    on RESERVATION
    for each row
declare
    places int;
    status char(1);
begin
    select NO_AVAILABLE_PLACES into places from TRIPS where TRIPS.TRIP_ID
= :new.TRIP_ID;
    select STATUS into status from RESERVATION where RESERVATION_ID =
:new.RESERVATION_ID;

    if status = 'c' and :new.STATUS = 'n' then
        if :new.NO_PLACES < places then
            raise_application_error(-20001, 'trip doest not have enough
available places');
        end if;
    end if;
end;
```

TR_UPDATE_TRIP_PLACES

```
create trigger TR_UPDATE_TRIP_PLACES
```

```
   before update
   on TRIP
   for each row
declare
   available_places int;
   old_max_places int;
begin
   select NO_AVAILABLE_PLACES into available_places from TRIPS where
TRIPS.TRIP_ID = :new.TRIP_ID;
   select MAX_NO_PLACES into old_max_places  from TRIPS where
TRIPS.TRIP_ID = :new.TRIP_ID;

   if :new.MAX_NO_PLACES + available_places < old_max_places then
       raise_application_error(-20001, 'trip does not have enough
available places');
   end if;

end;
```