

Raport z zajęć Oracle PL/SQL

Bartłomiej Plewnia

gr.wtorek 11.15 - 12.45 A

1. Tabele

1.1 Osoby

```
create table OSOBY(  
    ID_OSOBY INT generated always as identity not null,  
    imie varchar2(50),  
    nazwisko varchar2(50),  
    pesel varchar2(11),  
    kontakt varchar2(100),  
    constraint OSOBY_PK primary key(  
        ID_OSOBY  
    )  
    enable  
);
```

1.2 Wycieczki

```
create table WYCIECZKI  
(  
    ID_WYCIECZKI int generated always as identity not null,  
    nazwa varchar2(100),  
    kraj varchar2(50),  
    data date,  
    opis varchar2(200),  
    liczba_miejsc int,  
    constraint WYCIECZKI_PK primary key (ID_WYCIECZKI) enable  
);
```

1.3 Rezerwacje

```
create table REZERWACJE
(
    NR_REZERWACJI int generated always as identity not null,
    ID_WYCIECZKI int,
    ID_OSOBY int,
    status char(1),
    constraint REZERWACJE_PK primary key (NR_REZERWACJI) enable
);
```

1.4 Constrainty

```
alter table REZERWACJE
add constraint REZERWACJE_FK1 FOREIGN KEY(ID_OSOBY)
REFERENCES OSOBY (ID_OSOBY)
enable;
```

```
alter table REZERWACJE
add constraint REZERWACJE_FK2 foreign key (ID_WYCIECZKI)
REFERENCES WYCIECZKI(ID_WYCIECZKI)
enable;
```

```
alter table REZERWACJE
add constraint REZERWACJE_CHK1 check ( status in ('N','P','Z','A') )
enable;
```

2. Przykładowe dane

```
INSERT INTO OSOBY (imie, nazwisko, pesel, kontakt)
VALUES('Adam', 'Kowalski', '87654321', 'tel: 6623');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Jonas', 'Brother', '87623321', 'tel: 909321245');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Jan', 'Nowakowski', '12345678', 'tel: 23122987, prosze nie dzwonic');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Bartłomiej', 'Random', '99091257289', 'konak@gmail.com');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Jakub', 'Plesiak', '781214', 'linkedin to moja pasja');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Krzysztof', 'Jankowki', '00113005378', 'orangutan@onet.pl');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Szymon', 'Kowalski', '90123112345', 'ugabuga@gmail.com');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Adrian', 'Tumulec', '81293291923', '606-182-123');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Hubert', 'Hubart', '91029302034', 'koranadwora@wp.pl');
```

```
INSERT INTO osoby (imie, nazwisko, pesel, kontakt)
VALUES('Anna', 'Ozak', '9120392356', 'iaks@gmail.com');
commit work;
```

```
INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba_miejsc)
VALUES ('Wycieczka do Paryza', 'Francja', TO_DATE('2016-01-01', 'YYYY-MM-DD'), 'Nic nie warta
wycieczka', 120);
```

```
INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba_miejsc)
VALUES ('Wycieczka krajoznawcza do Poznania', 'Polska', TO_DATE('2021-01-01',
'YYYY-MM-DD'), 'Osobiscie polecam', 20);
```

```
INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba_miejsc)
VALUES ('Trip to Los Angeles', 'USA', TO_DATE('2018-12-31', 'YYYY-MM-DD'), 'Great
experience', 12);
```

```
INSERT INTO wycieczki (nazwa, kraj, data, opis, liczba_miejsc)
VALUES ('Wycieczka do Berlina', 'Niemcy', TO_DATE('2025-08-21', 'YYYY-MM-DD'), 'Pisal od niej
Sienkiewicz', 40);
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (41,56,'A');
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (42,57,'Z');
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (43,58,'N');
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (44,59,'A');
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (41,60,'Z');
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (42,61,'P');
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (43,62,'N');
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (44,63,'P');
```





```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (41,64,'P');
```

```
INSERT into REZERWACJE (ID_WYCIECZKI, ID_OSOBY, status)
VALUES (42,65,'A');
commit work;
```

Po wykonaniu powyższych poleceń table wypełniły się danymi.

	ID_OSOBY	IMIE	NAZWISKO	PESEL	KONTAKT
1	56	Adam	Kowalski	87654321	tel: 6623
2	57	Jonas	Brother	87623321	tel: 909321245
3	58	Jan	Nowakowski	12345678	tel: 23122987, prosze nie dzwonic
4	59	Bartłomiej	Random	99091257289	konak@gmail.com
5	60	Jakub	Plesiak	781214	linkedin to moja pasja
6	61	Krzysztof	Jankowski	00113005378	orangutan@onet.pl
7	62	Szymon	Kowalski	90123112345	ugabuga@gmail.com
8	63	Adrian	Tumulec	81293291923	606-182-123
9	64	Hubert	Hubart	91029302034	koranadwora@wp.pl
10	65	Anna	Ozak	9120392356	iaks@gmail.com

	ID_WYCIECZKI	NAZWA	KRAJ	DATA	OPIS	LICZBA_MIEJSC
1	41	Wycieczka do Paryza	Francja	2016-01-01 00:00:00	Nic nie warta wycieczka	120
2	42	Wycieczka krajoznawcza do Poznania	Polska	2021-01-01 00:00:00	Osobiscie polecam	20
3	43	Trip to Los Angeles	USA	2018-12-31 00:00:00	Great experience	12
4	44	Wycieczka do Berlina	Niemcy	2025-08-21 00:00:00	Pisal od niej Sienkiewicz	40

	 NR_REZERWACJI ▾	 ID_WYCIECZKI ▾	 ID_OSOBY ▾	 STATUS ▾
1	61	41	56	A
2	62	42	57	Z
3	63	43	58	N
4	64	44	59	A
5	65	41	60	Z
6	66	42	61	P
7	67	43	62	N
8	68	44	63	P
9	69	41	64	P
10	70	42	65	A

3. Widoki

3.1 All Reservations

```
create view AllReservations as
select w.kraj, w.data, w.nazwa, o.imie, o.nazwisko, r.status
from WYCIECZKI w inner join REZERWACJE r on r.ID_WYCIECZKI = w.ID_WYCIECZKI
inner join OSOBY O on r.ID_OSOBY = O.ID_OSOBY;
```

	KRAJ	DATA	NAZWA	IMIE	NAZWISKO	STATUS
1	Francja	2016-01-01 00:00:00	Wycieczka do Paryza	Adam	Kowalski	A
2	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	Jonas	Brother	Z
3	USA	2018-12-31 00:00:00	Trip to Los Angeles	Jan	Nowakowski	N
4	Niemcy	2025-08-21 00:00:00	Wycieczka do Berlina	Bartłomiej	Random	A
5	Francja	2016-01-01 00:00:00	Wycieczka do Paryza	Jakub	Plesiak	Z
6	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	Krzysztof	Jankowki	P
7	USA	2018-12-31 00:00:00	Trip to Los Angeles	Szymon	Kowalski	N
8	Niemcy	2025-08-21 00:00:00	Wycieczka do Berlina	Adrian	Tumulec	P
9	Francja	2016-01-01 00:00:00	Wycieczka do Paryza	Hubert	Hubart	P
10	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	Anna	Ozak	A

3.2 Accepted Reservations

```
create view AcceptedReservations as
select w.kraj, w.data, w.nazwa, o.imie, o.nazwisko, r.status
from WYCIECZKI w inner join REZERWACJE r on r.ID_WYCIECZKI = w.ID_WYCIECZKI
inner join OSOBY O on r.ID_OSOBY = O.ID_OSOBY
where r.status = 'P';
```

	KRAJ	DATA	NAZWA	IMIE	NAZWISKO	STATUS
1	Francja	2016-01-01 00:00:00	Wycieczka do Paryza	Hubert	Hubart	P
2	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	Krzysztof	Jankowki	P
3	Niemcy	2025-08-21 00:00:00	Wycieczka do Berlina	Adrian	Tumulec	P

3.3 Reservations in future

```
create view ReservationsInFuture as
select w.kraj, w.data, w.nazwa, o.imie, o.nazwisko, r.status
from WYCIECZKI w inner join REZERWACJE r on r.ID_WYCIECZKI = w.ID_WYCIECZKI
inner join OSOBY O on r.ID_OSOBY = O.ID_OSOBY
where w.data - current_date > 0;
```

	KRAJ	DATA	NAZWA	IMIE	NAZWISKO	STATUS
1	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	Jonas	Brother	Z
2	Niemcy	2025-08-21 00:00:00	Wycieczka do Berlina	Bartłomiej	Random	A
3	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	Krzysztof	Jankowki	P
4	Niemcy	2025-08-21 00:00:00	Wycieczka do Berlina	Adrian	Tumulec	P
5	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	Anna	Ozak	A

3.4 Places of Trips

create view PlacesOfTrips **as**

```
select w.kraj, w.data, w.nazwa, w.liczba_miejsc,  
       w.liczba_miejsc - (select count(*) from REZERWACJE r where r.ID_WYCIECZKI =  
w.ID_WYCIECZKI ) FreeSeats  
from WYCIECZKI w;
```

	KRAJ	DATA	NAZWA	LICZBA_MIEJSC	FREESEATS
1	Francja	2016-01-01 00:00:00	Wycieczka do Paryza	120	117
2	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	20	17
3	USA	2018-12-31 00:00:00	Trip to Los Angeles	12	10
4	Niemcy	2025-08-21 00:00:00	Wycieczka do Berlina	40	38

3.5 Available Trips

create view AvailableTrips **as**

```
select w.ID_WYCIECZKI, w.kraj, w.data, w.nazwa, w.liczba_miejsc,  
       w.liczba_miejsc - (select count(*) from REZERWACJE r where r.ID_WYCIECZKI =  
w.ID_WYCIECZKI ) FreeSeats  
from WYCIECZKI w  
where w.data - current_date > 0  
and w.liczba_miejsc - (select count(*) from REZERWACJE r where r.ID_WYCIECZKI =  
w.ID_WYCIECZKI ) > 0;
```

	ID_WYCIECZKI	KRAJ	DATA	NAZWA	LICZBA_MIEJSC	FREESEATS
1	42	Polska	2021-01-01 00:00:00	Wycieczka krajoznawcza do Poznania	20	17
2	44	Niemcy	2025-08-21 00:00:00	Wycieczka do Berlina	40	38

4. Funkcje pobierające dane

Do prawidłowego funkcjonowania pewnych funkcji potrzebne było stworzenie swoich obiektów.

Obiekt: PersonAndTrip

CREATE OR REPLACE TYPE PersonAndTrip **AS OBJECT**

```
(  
  NAZWA   NVARCHAR2(100),  
  KRAJ    NVARCHAR2(50),  
  "DATA"  DATE,  
  IMIE    NVARCHAR2(50),  
  NAZWISKO NVARCHAR2(50),  
  STATUS  CHAR(1)  
);
```

CREATE OR REPLACE TYPE PersonAndTrip_Table **IS TABLE OF** PersonAndTrip;

Obiekt: Trip

CREATE OR REPLACE TYPE Trip **AS OBJECT**

```
(  
  NAZWA          VARCHAR2(100),  
  KRAJ           VARCHAR2(50),  
  "DATA"         DATE,  
  OPIS           VARCHAR2(200),  
  LICZBA_MIEJSC  NUMBER  
);
```

CREATE OR REPLACE TYPE Trip_Table **IS TABLE OF** Trip;

4.1 Trip Attendees

CREATE OR REPLACE FUNCTION TripAttendees(id WYCIECZKI.ID_WYCIECZKI%TYPE)

RETURN PersonAndTrip_Table

AS

result PersonAndTrip_Table;

trip_exists **INT**;

BEGIN

SELECT COUNT(*) **INTO** trip_exists **FROM** WYCIECZKI **WHERE** WYCIECZKI.ID_WYCIECZKI =
TripAttendees.id;

IF trip_exists = 0 **THEN**

RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID not found.');

END IF;

SELECT w.NAZWA, w.KRAJ, w.DATA, o.IMIE, o.NAZWISKO, r.STATUS


```

INTO result
FROM WYCIECZKI w
    JOIN REZERWACJE r
        ON w.ID_WYCIECZKI = r.ID_WYCIECZKI
    JOIN OSOBY o
        ON r.ID_OSOBY = o.ID_OSOBY
WHERE w.ID_WYCIECZKI = id
    AND r.STATUS <> 'A';

RETURN result;
END;

```

4.2 Trips of a person

```

CREATE OR REPLACE FUNCTION TripsOfPerson(id OSOBY.ID_OSOBY%TYPE)
    RETURN PersonAndTrip_Table
AS
    result    PersonAndTrip_Table;
    person_exists INT;
BEGIN
    SELECT COUNT(*) INTO person_exists FROM OSOBY WHERE OSOBY.ID_OSOBY =
TripsOfPerson.id;

    IF person_exists = 0 THEN
        RAISE_APPLICATION_ERROR(-20000, 'Person with chosen ID not found.');
```

```

    END IF;

    SELECT w.NAZWA, w.KRAJ, w.DATA, o.IMIE, o.NAZWISKO, r.STATUS
INTO result
FROM WYCIECZKI w
    JOIN REZERWACJE r
        ON w.ID_WYCIECZKI = r.ID_WYCIECZKI
    JOIN OSOBY o
        ON r.ID_OSOBY = o.ID_OSOBY
WHERE o.ID_OSOBY = id
    AND r.STATUS <> 'A';

    RETURN result;
END;

```

4.3 Open Trips

```

CREATE OR REPLACE FUNCTION OpenTrips(Country WYCIECZKI.kraj%TYPE, StartDate
WYCIECZKI.data%TYPE, EndDate WYCIECZKI.data%TYPE)
    RETURN Trip_Table
AS
    result    Trip_Table;
    country_exists int;

```

```

on_target int;
BEGIN
  SELECT COUNT(*) INTO country_exists FROM WYCIECZKI WHERE
WYCIECZKI.ID_WYCIECZKI = OpenTrips.Country;

  IF country_exists = 0 THEN
    RAISE_APPLICATION_ERROR(-20000, 'There is no trip to that country at all');
  END IF;

  SELECT COUNT(*) INTO on_target FROM WYCIECZKI WHERE WYCIECZKI.ID_WYCIECZKI =
OpenTrips.Country and WYCIECZKI.data between StartDate and EndDate;

  IF on_target = 0 THEN
    RAISE_APPLICATION_ERROR(-20000, 'There is no trip to that country in that term');
  END IF;

  SELECT w.NAZWA, w.KRAJ, w.DATA, w.opis, w.liczba_miejsc
  INTO result
  FROM WYCIECZKI w
  where w.ID_WYCIECZKI = OpenTrips.Country and w.data between StartDate and EndDate

  RETURN result;
END;

```

4.4 Free Places - pomocnicza funkcja stworzona w celu uniknięcia powielania kodu

```

create or replace function FreePlaces(TripID WYCIECZKI.ID_WYCIECZKI%TYPE)
  return int
as
  result int;
  numberOfReservations int;
begin
  select count(*)
  into numberOfReservations
  from REZERWACJE r
  where r.ID_WYCIECZKI = FreePlaces.TripID;

  select w.liczba_miejsc - numberOfReservations
  into result
  from WYCIECZKI w
  where w.ID_WYCIECZKI = TripID;

  return result;
end;

```

5. Procedury modyfikujące dane

W procedurach jest już zawarte dopisywanie do tabeli REZERWACJE_LOG z punktu 6

5.1 Add Reservation

CREATE OR REPLACE PROCEDURE

AddReservation (TripID WYCIECZKI.ID_WYCIECZKI%TYPE,
PersonID OSOBY.ID_OSOBY%TYPE)

AS

person_exists INT;
trip_available INT;
reservation_exists INT;
new_reservation_ID INT;

BEGIN

SELECT COUNT(*) INTO person_exists FROM OSOBY WHERE OSOBY.ID_OSOBY =
AddReservation.PersonID;

IF person_exists = 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Person with chosen ID not found.');

END IF;

SELECT COUNT(*)

INTO trip_available

FROM AvailableTrips2

WHERE AvailableTrips2.ID_WYCIECZKI = *AddReservation*.TripID;

IF trip_available = 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID not available.');

END IF;

SELECT COUNT(*)

INTO reservation_exists

FROM REZERWACJE

WHERE REZERWACJE.ID_WYCIECZKI = *AddReservation*.TripID

AND REZERWACJE.ID_OSOBY = *AddReservation*.PersonID;

IF reservation_exists > 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Reservation for chosen trip ID and person ID
already exists.');

END IF;

INSERT INTO REZERWACJE (ID_WYCIECZKI, ID_OSOBY, STATUS)

VALUES (*AddReservation*.TripID, *AddReservation*.PersonID, 'N')

RETURNING NR_REZERWACJI INTO new_reservation_ID;

INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)

```

VALUES (new_reservation_ID, CURRENT_DATE, 'N');

COMMIT;
END;

begin
    ADDRESERVATION( TripID: 44, PersonID: 64 );
end;

select * from REZERWACJE;

```

	NR_REZERWACJI	ID_WYCIECZKI	ID_OSOBY	STATUS
9	69	41	64	P
10	70	42	65	A
11	122	42	64	N
12	123	44	64	N

Została dodana rezerwacja o numerze 123.

5.2 Change status of Reservation

CREATE OR REPLACE PROCEDURE

ChangeStatusOfReservation(ReservationID REZERWACJE.NR_REZERWACJI%TYPE,
Status REZERWACJE.status%TYPE)

AS

```

reservation_exists INT;
TripID WYCIECZKI.ID_WYCIECZKI%TYPE;
current_status char(1);

```

BEGIN

```

SELECT COUNT(*)
INTO reservation_exists
FROM REZERWACJE
WHERE REZERWACJE.NR_REZERWACJI = ChangeStatusOfReservation.ReservationID;

```

```

IF reservation_exists = 0 THEN

```

```

    RAISE_APPLICATION_ERROR(-20000, 'Reservation does not exist.');
```

```

END IF;

```

```

select status

```

```

into current_status

```

```

from REZERWACJE

```

```

    where ReservationID = NR_REZERWACJI;

```

```

IF current_status = ChangeStatusOfReservation.Status then

```

```

    RAISE_APPLICATION_ERROR(-20000, 'The desirable status is already there.');
```

```

end if;

```

```

select ID_WYCIECZKI
into TripID
from REZERWACJE
where NR_REZERWACJI = ReservationID;

IF FreePlaces(TripID) = 0 and current_status = 'A' then
    RAISE_APPLICATION_ERROR(-20000, 'There is no more places left for that trip.');
```

```

end if;

UPDATE REZERWACJE r
SET r.STATUS = ChangeStatusOfReservation.Status
where NR_REZERWACJI = ChangeStatusOfReservation.ReservationID;
commit;

INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
VALUES (ChangeStatusOfReservation.ReservationID, CURRENT_DATE,
ChangeStatusOfReservation.Status);

COMMIT;
END;
begin
    CHANGESTATUSOFRESERVATION( ReservationID: 123, Status: 'P' );
end;
select * from REZERWACJE;
```

	NR_REZERWACJI	ID_WYCIECZKI	ID_OSOBY	STATUS
9	69	41	64	P
10	70	42	65	A
11	122	42	64	N
12	123	44	64	P

Rezerwacja o numerze 123 została zmieniona na status 'P'.

5.3 Change number of places

```

CREATE OR REPLACE PROCEDURE ChangeNumberOfPlaces(id
WYCIECZKI.ID_WYCIECZKI%TYPE,
DesirableNumberOfPlaces WYCIECZKI.liczba_miejsc%TYPE)
AS
    TripID WYCIECZKI.ID_WYCIECZKI%TYPE;
    NumberOfPlaces int;
BEGIN
    select count(*)
    into TripID
    from WYCIECZKI w
    where w.ID_WYCIECZKI = ChangeNumberOfPlaces.id;

    if TripID = 0 then
        RAISE_APPLICATION_ERROR(-20000, 'The is no such trip.');
```

```

    end if;
```

```

select w.liczba_miejsc
into NumberOfPlaces
from WYCIECZKI w
where w.ID_WYCIECZKI = ChangeNumberOfPlaces.id;

if ChangeNumberOfPlaces.DesirableNumberOfPlaces < NumberOfPlaces - FreePlaces(id) then
    RAISE_APPLICATION_ERROR(-20000, 'There is already reserved more places. ');
end if;

UPDATE WYCIECZKI w
set w.liczba_miejsc = DesirableNumberOfPlaces
where w.ID_WYCIECZKI = id;
commit;
END;

begin
    ChangeNumberOfPlaces( id: 42, DesirableNumberOfPlaces: 100 );
end;
select * from WYCIECZKI;

```

ID_WYCIECZKI	NAZWA	KRAJ	DATA	OPIS	LICZBA_MIEJSC	LICZBA_WOLNYCH_MIEJSC
1	41 Wycieczka do <small>NAZWA: VARCHAR2(100)</small>	Francja	2016-01-01 00:00:00	Nic nie warta wycieczka	120	118
2	42 Wycieczka krajoznawcza do Poznania	Polska	2021-01-01 00:00:00	Osobiscie polecam	100	17

6. Dodanie tabeli Rezerwacje_log

```

CREATE TABLE REZERWACJE_LOG
(
    ID                INT GENERATED ALWAYS AS IDENTITY NOT NULL,
    ID_REZERWACJI INT,
    DATA             DATE,
    STATUS            CHAR(1),
    CONSTRAINT REZERWACJE_LOG_PK PRIMARY KEY (ID) ENABLE
);

```

7. Dodanie redundantnego pola

liczba_wolnych_miejsc oraz zmiana procedur, funkcji oraz widoków.

7.1 Dodanie pola liczba_wolnych_miejsc

```
Alter table WYCIECZKI  
add liczba_wolnych_miejsc int;
```

7.2 Modyfikacja widoków

```
create view AllReservations2 as  
select w.kraj, w.data, w.nazwa, w.liczba_wolnych_miejsc, o.imie, o.nazwisko, r.status  
from WYCIECZKI w inner join REZERWACJE r on r.ID_WYCIECZKI = w.ID_WYCIECZKI  
inner join OSOBY O on r.ID_OSOBY = O.ID_OSOBY;
```

```
create view AcceptedReservations2 as  
select w.kraj, w.data, w.nazwa, w.liczba_wolnych_miejsc, o.imie, o.nazwisko, r.status  
from WYCIECZKI w inner join REZERWACJE r on r.ID_WYCIECZKI = w.ID_WYCIECZKI  
inner join OSOBY O on r.ID_OSOBY = O.ID_OSOBY  
where r.status = 'P';
```

```
create view ReservationsInFuture2 as  
select w.kraj, w.data, w.nazwa, w.liczba_wolnych_miejsc, o.imie, o.nazwisko, r.status  
from WYCIECZKI w inner join REZERWACJE r on r.ID_WYCIECZKI = w.ID_WYCIECZKI  
inner join OSOBY O on r.ID_OSOBY = O.ID_OSOBY  
where w.data - current_date > 0;
```

```
create view PlacesOfTrips2 as  
select w.kraj, w.data, w.nazwa, w.liczba_wolnych_miejsc, w.liczba_miejsc  
from WYCIECZKI w;
```

```
create view AvailableTrips2 as  
select w.ID_WYCIECZKI, w.kraj, w.data, w.nazwa, w.liczba_wolnych_miejsc, w.liczba_miejsc  
from WYCIECZKI w  
where w.data - current_date > 0  
and w.liczba_wolnych_miejsc > 0;
```


7.3 Modyfikacja procedur

CREATE OR REPLACE PROCEDURE

AddReservation2(TripID WYCIECZKI.ID_WYCIECZKI%TYPE,
PersonID OSOBY.ID_OSOBY%TYPE)

AS

person_exists INT;
trip_available INT;
reservation_exists INT;
new_reservation_ID INT;

BEGIN

SELECT COUNT(*) INTO person_exists FROM OSOBY WHERE OSOBY.ID_OSOBY =
AddReservation2.PersonID;

IF person_exists = 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Person with chosen ID not found.');

END IF;

SELECT COUNT(*)

INTO trip_available

FROM AvailableTrips2

WHERE AvailableTrips2.ID_WYCIECZKI = *AddReservation2*.TripID;

IF trip_available = 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID not available.');

END IF;

SELECT COUNT(*)

INTO reservation_exists

FROM REZERWACJE

WHERE REZERWACJE.ID_WYCIECZKI = *AddReservation2*.TripID

AND REZERWACJE.ID_OSOBY = *AddReservation2*.PersonID;

IF reservation_exists > 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Reservation for chosen trip ID and person ID
already exists.');

END IF;

INSERT INTO REZERWACJE (ID_WYCIECZKI, ID_OSOBY, STATUS)

VALUES (*AddReservation2*.TripID, *AddReservation2*.PersonID, 'N')

RETURNING NR_REZERWACJI INTO new_reservation_ID;

begin

COUNTFREEPLACES();

end;

INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)

VALUES (new_reservation_ID, CURRENT_DATE, 'N');

COMMIT;

```
END;  
commit work;
```

CREATE OR REPLACE PROCEDURE

```
ChangeStatusOfReservation2(ReservationID REZERWACJE.NR_REZERWACJI%TYPE,  
    Status REZERWACJE.status%TYPE)
```

AS

```
reservation_exists INT;  
TripID WYCIECZKI.ID_WYCIECZKI%TYPE;  
current_status char(1);
```

BEGIN

```
SELECT COUNT(*)  
INTO reservation_exists  
FROM REZERWACJE  
WHERE REZERWACJE.NR_REZERWACJI = ChangeStatusOfReservation2.ReservationID;
```

```
IF reservation_exists = 0 THEN  
    RAISE_APPLICATION_ERROR(-20000, 'Reservation does not exist.');
```

```
END IF;
```

--status introduced is the same

```
select status  
into current_status  
from REZERWACJE  
where ReservationID = NR_REZERWACJI;
```

```
IF current_status = ChangeStatusOfReservation2.Status then  
    RAISE_APPLICATION_ERROR(-20000, 'The desirable status is already there.');
```

```
end if;
```

```
select ID_WYCIECZKI  
into TripID  
from REZERWACJE  
where NR_REZERWACJI = ReservationID;
```

```
IF FreePlaces(TripID) = 0 and current_status = 'A' then  
    RAISE_APPLICATION_ERROR(-20000, 'There is no more places left for that trip.');
```

```
end if;
```

```
UPDATE REZERWACJE r  
SET r.STATUS = ChangeStatusOfReservation2.Status  
where NR_REZERWACJI = ChangeStatusOfReservation2.ReservationID;  
commit;
```

```
INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)  
VALUES (ChangeStatusOfReservation2.ReservationID, CURRENT_DATE,  
ChangeStatusOfReservation2.Status);
```

```
COMMIT;  
END;  
commit work;
```

```
CREATE OR REPLACE PROCEDURE ChangeNumberOfPlaces2(id  
WYCIECZKI.ID_WYCIECZKI%TYPE, DesirableNumberOfPlaces  
WYCIECZKI.liczba_miejsc%TYPE)
```

```
AS
```

```
    TripID WYCIECZKI.ID_WYCIECZKI%TYPE;  
    NumberOfPlaces int;  
    NumberOfFreePlaces int;
```

```
BEGIN
```

```
    select count(*)  
    into TripID  
    from WYCIECZKI w  
    where w.ID_WYCIECZKI = ChangeNumberOfPlaces2.id;
```

```
    if TripID = 0 then  
        RAISE_APPLICATION_ERROR(-20000, 'The is no such trip.');
```

```
    end if;
```

```
    select w.liczba_miejsc  
    into NumberOfPlaces  
    from WYCIECZKI w  
    where w.ID_WYCIECZKI = ChangeNumberOfPlaces2.id;
```

```
    select w.liczba_wolnych_miejsc  
    into NumberOfFreePlaces  
    from WYCIECZKI w  
    where w.ID_WYCIECZKI = ChangeNumberOfPlaces2.id;
```

```
    if ChangeNumberOfPlaces2.DesirableNumberOfPlaces < NumberOfPlaces - NumberOfFreePlaces  
    then
```

```
        RAISE_APPLICATION_ERROR(-20000, 'There is already reserved more places.')
```

```
    end if;
```

```
    UPDATE WYCIECZKI w  
    set w.liczba_miejsc = DesirableNumberOfPlaces  
    where w.ID_WYCIECZKI = id;  
    commit;
```

```
COMMIT;  
END;  
commit work;
```

8. Zapisywanie do dziennika Rezerwacje_log przy pomocy triggerów

8.1 Add Reservation Trigger

```
CREATE OR REPLACE TRIGGER AddReservationTrigger
AFTER INSERT
ON REZERWACJE
FOR EACH ROW
BEGIN
  INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
  VALUES (:NEW.NR_REZERWACJI, CURRENT_DATE, :NEW.STATUS);

  --adding reservation will always remove one free place from Trip
  UPDATE WYCIECZKI w
  SET w.liczba_wolnych_miejsc = w.liczba_wolnych_miejsc - 1
  WHERE w.ID_WYCIECZKI = :NEW.ID_WYCIECZKI;
END;
```

8.2 Change Status Trigger

```
CREATE OR REPLACE TRIGGER ChangeStatusTrigger
AFTER INSERT
ON REZERWACJE
FOR EACH ROW
declare
  new_places int;
BEGIN
  INSERT INTO REZERWACJE_LOG (ID_REZERWACJI, DATA, STATUS)
  VALUES (:NEW.NR_REZERWACJI, current_date, :NEW.STATUS);

  --when changing status from 'A' to any other there will be one additional place
  if :NEW.STATUS = 'A' then
    new_places := 1;
  else
    new_places := 0;
  end if;

  UPDATE WYCIECZKI w
  SET w.liczba_wolnych_miejsc = w.liczba_wolnych_miejsc + new_places
  WHERE w.ID_WYCIECZKI = :NEW.ID_WYCIECZKI;
END;
```

8.3 Block Deletion of Reservation Trigger

```
REATE OR REPLACE TRIGGER BlockDeletionOfReservationTrigger
BEFORE DELETE
ON REZERWACJE
```

```
FOR EACH ROW
BEGIN
    RAISE_APPLICATION_ERROR(-20000, 'Deleting reservation is forbidden.');
```

END;

8.4 Changing Number of Places Trigger

```
CREATE OR REPLACE TRIGGER ChangingNumberOfPlacesTrigger
AFTER UPDATE OF liczba_miejsc
ON WYCIECZKI
FOR EACH ROW
BEGIN
    Update WYCIECZKI
    set liczba_miejsc = liczba_miejsc + (:NEW.liczba_miejsc - liczba_miejsc);
END;
```

9. Modyfikacja procedur tak aby współdziałały z triggerami z poprzedniego punktu

CREATE OR REPLACE PROCEDURE

AddReservation3(TripID WYCIECZKI.ID_WYCIECZKI%TYPE,
PersonID OSOBY.ID_OSOBY%TYPE)

AS

person_exists INT;
trip_available INT;
reservation_exists INT;
new_reservation_ID INT;

BEGIN

SELECT COUNT(*) INTO person_exists FROM OSOBY WHERE OSOBY.ID_OSOBY =
AddReservation3.PersonID;

IF person_exists = 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Person with chosen ID not found.');

END IF;

SELECT COUNT(*)

INTO trip_available

FROM AvailableTrips

WHERE AvailableTrips.ID_WYCIECZKI = *AddReservation3*.TripID;

IF trip_available = 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Trip with chosen ID not available.');

END IF;

SELECT COUNT(*)

INTO reservation_exists

FROM REZERWACJE

WHERE REZERWACJE.ID_WYCIECZKI = *AddReservation3*.TripID

AND REZERWACJE.ID_OSOBY = *AddReservation3*.PersonID;

IF reservation_exists > 0 THEN

RAISE_APPLICATION_ERROR(-20000, 'Reservation for chosen trip ID and person ID
already exists.');

END IF;

INSERT INTO REZERWACJE (ID_WYCIECZKI, ID_OSOBY, STATUS)

VALUES (*AddReservation3*.TripID, *AddReservation3*.PersonID, 'N')

RETURNING NR_REZERWACJI INTO new_reservation_ID;

COMMIT;

END;

commit work;

CREATE OR REPLACE PROCEDURE

ChangeStatusOfReservation3(ReservationID REZERWACJE.NR_REZERWACJI%TYPE,
Status REZERWACJE.status%TYPE)

AS

reservation_exists INT;
TripID WYCIECZKI.ID_WYCIECZKI%TYPE;
current_status char(1);

BEGIN

SELECT COUNT(*)
INTO reservation_exists
FROM REZERWACJE
WHERE REZERWACJE.NR_REZERWACJI = *ChangeStatusOfReservation3*.ReservationID;

IF reservation_exists = 0 THEN
RAISE_APPLICATION_ERROR(-20000, 'Reservation does not exist.');

END IF;

select status
into current_status
from REZERWACJE
where ReservationID = NR_REZERWACJI;

IF current_status = *ChangeStatusOfReservation3*.Status then
RAISE_APPLICATION_ERROR(-20000, 'The desirable status is already there.');

end if;

select ID_WYCIECZKI
into TripID
from REZERWACJE
where NR_REZERWACJI = ReservationID;

IF *FreePlaces*(TripID) = 0 and current_status = 'A' then
RAISE_APPLICATION_ERROR(-20000, 'There is no more places left for that trip.');

end if;

UPDATE REZERWACJE r
SET r.STATUS = *ChangeStatusOfReservation3*.Status
where NR_REZERWACJI = *ChangeStatusOfReservation3*.ReservationID;

--everything else is in the trigger

COMMIT;

END;

commit work;

CREATE OR REPLACE PROCEDURE *ChangeNumberOfPlaces3*(id

WYCIECZKI.ID_WYCIECZKI%TYPE, DesirableNumberOfPlaces
WYCIECZKI.liczba_miejsc%TYPE)

AS

TripID WYCIECZKI.ID_WYCIECZKI%TYPE;


```
NumberOfPlaces int;  
BEGIN  
  select count(*)  
  into TripID  
  from WYCIECZKI w  
  where w.ID_WYCIECZKI = ChangeNumberOfPlaces3.id;  
  
  if TripID = 0 then  
    RAISE_APPLICATION_ERROR(-20000, 'The is no such trip.');
```

```
  end if;  
  
  select w.liczba_miejsc  
  into NumberOfPlaces  
  from WYCIECZKI w  
  where w.ID_WYCIECZKI = ChangeNumberOfPlaces3.id;  
  
  if ChangeNumberOfPlaces3.DesirableNumberOfPlaces < NumberOfPlaces - FreePlaces(id) then  
    RAISE_APPLICATION_ERROR(-20000, 'There is already reserved more places.');
```

```
  end if;  
  
  UPDATE WYCIECZKI w  
  set w.liczba_miejsc = DesirableNumberOfPlaces  
  where w.ID_WYCIECZKI = id;  
  
  COMMIT;  
END;  
  commit work;
```