

## Oracle PL/SQL,- ćwiczenie

### PL/SQL – programowanie proceduralne, widoki, procedury, triggery

#### 1. Tabele

Trip (trip\_id, name, country, date, max\_no\_places)

Person(person\_id, firstname, lastname)

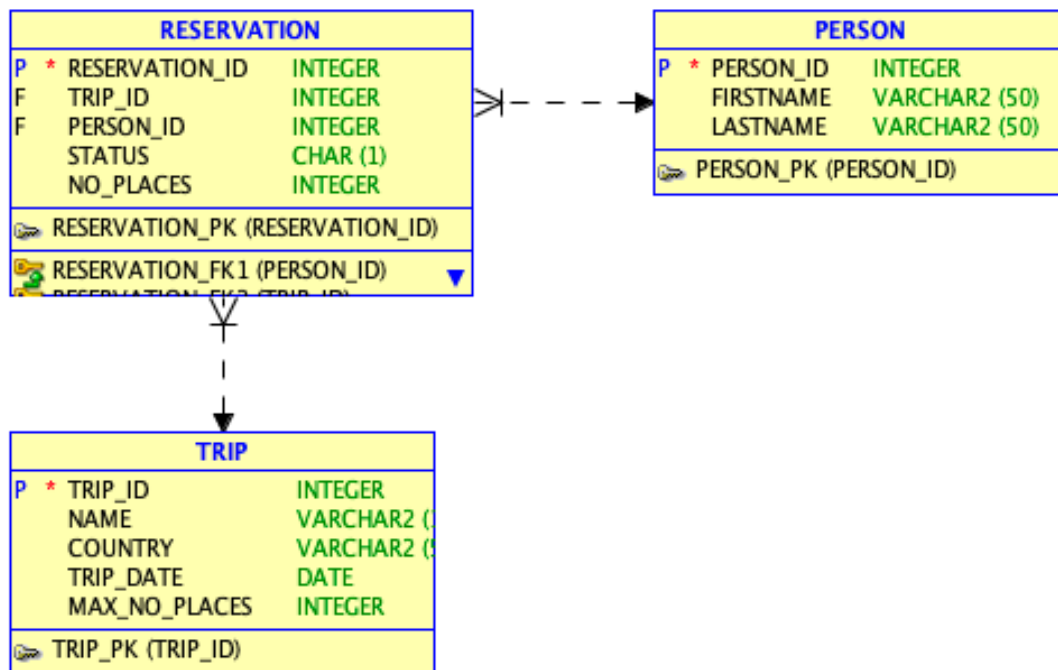
Reservation(reservation\_id, trip\_id, person\_id, status,no\_places)

Pole status w tabeli Rezerwacje może przyjmować jedną z 4 wartości

N – New

P – Paid

C – Canceled



## Tworzenie tabel.

```
--tables
create table person
(
    person_id int generated always as identity not null
,   firstname varchar2(50)
,   lastname  varchar2(50)
,   constraint person_pk primary key
    (
        person_id
    )
    enable
);

create table trip
(
    trip_id int generated always as identity not null
,   name varchar2(100)
,   country varchar2(50)
,   trip_date date
,   max_no_places int
,   constraint trip_pk primary key
    (
        trip_id
    )
    enable
);

create table reservation
(
    reservation_id int generated always as identity not null
,   trip_id int
,   person_id int
,   status char(1)
,   no_places int
,   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. Wypełnianie tabeli przykładowymi danymi
  - 4 wycieczki
  - 10 osób
  - 10 rezerwacji

Dane testowe powinny być różnorodne (wycieczki w przyszłości, wycieczki w przeszłości, rezerwacje o różnym statusie itp.) tak, żeby umożliwić testowanie napisanych procedur. W razie potrzeby należy zmodyfikować dane tak żeby przetestować różne przypadki.

```

insert into person (firstname, lastname)
values('adam', 'kowalski');

insert into person (firstname, lastname)
values('jan', 'nowak');

select * from person;

insert into trip (name, country, trip_date, max_no_places)
values ('wycieczka do paryza','francja',to_date('2021-09-03','yyyy-mm-dd'),3);

insert into trip (name, country, trip_date, max_no_places)
values ('wycieczka do krakowa','polska',to_date('2022-12-05','yyyy-mm-dd'),2);

select * from trip;

insert into reservation(trip_id, person_id, no_places, status)
values (1,1,1,'n');

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

```

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

- a) reservations(country,trip\_date,trip\_name, firstname, lastname,reservation\_id,no\_places,status)
- b) trips(country,trip\_date, trip\_name,max\_no\_places, no\_available\_places)
- c) available\_trips(country,trip\_date, trip\_name,no\_places, no\_available\_places)

Proponowany zestaw widoków można rozbudować wedle uznania/potrzeb. Można zmienić strukturę widoków, no dodając potrzebne pola/kolumny

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

- a) trip\_participants (trip\_id), procedura ma zwracać podobny zestaw danych jak widok reservations
- b) person\_reservations(person\_id), procedura ma zwracać podobny zestaw danych jak widok reservations
- c) available\_trips\_to(country, date\_from, date\_to)

Procedury/funkcje powinny zwracać tabelę/zbiór wynikowy

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

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

- a) add\_reservation(trip\_id, person\_id,no\_places), procedura powinna kontrolować czy wycieczka jeszcze się nie odbyła, i czy sa wolne miejsca
- b) modify\_reservation\_status(reservation\_id, status), procedura kontrolować czy możliwa jest zmiana statusu, np. zmiana statusu już anulowanej wycieczki (przywrócenie do stanu aktywnego nie zawsze jest możliwa – może już nie być miejsc)
- c) modify\_reservation(reservation\_id, no\_places), procedura kontrolować czy możliwa jest zmiana liczby zarezerwowanych miejsc
- d) modify\_max\_no\_places(trip\_id, no\_places), nie wszystkie zmiany liczby miejsc są dozwolone, nie można zmniejszyć liczby miejsc na wartość poniżej liczby zarezerwowanych miejsc

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

6. Dodajemy tabelę dziennikującą zmiany statusu rezerwacji, oraz liczby zarezerwowanych miejsc

log(id, reservation\_id, log\_date, status, no\_places)

Obsługa zapisywania inf. do dziennika/logu może być realizowana za pomocą procedur lub przy pomocy triggerów.

Należy zmodyfikować procedury dodające i zmieniające rezerwacje lub napisać odpowiednie triggerery

trigger obsługujący dodanie rezerwacji

trigger obsługujący zmianę statusu

trigger obsługujący zmianę liczby miejsc

7. Zmiana strategii obsługi kontroli dostępności miejsc. Operacje kontroli dostępności miejsc należy zrealizować przy pomocy triggerów (tzn operacje kontroli dostępności miejsc powinny zostać przeniesione z procedur do triggerów).

trigger obsługujący dodanie rezerwacji

trigger obsługujący zmianę statusu

trigger obsługujący zmianę liczby miejsc rezerwacji

trigger obsługujący zmianę maksymalnej liczby miejsc na poziomie wycieczki

## Podsumowanie

Należy przygotować raport z wykonania ćwiczenia. Raport powinien zawierać polecenia SQL (między innymi kod widoków, procedur, triggerów), wynik działania oraz krótki komentarz (jeśli jest potrzebny).

Raport należy przesłać w formie pliku PDF.

Dodatkowo należy też dołączyć kod zaimplementowanych widoków/procedur/triggerów w osobnym pliku tekstowym (.sql).

Nazwy plików:

nazwisko\_imie.pdf

nazwisko\_imie.sql

UWAGA: Proszę zwrócić uwagę na formatowanie kodu widoków/procedur/triggerów