AGH

AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE

Bazy danych

Laboratorium 1 - Oracle PL/SQL

Jan Gołda, WIET 2018/2019

Spis treści

Cały kod dostępny jest w repozytorium

https://github.com/regzand/AGH-Databases/tree/master/lab1

1. Tabele

1.1. Osoby

```
create table PERSONS
(
  PERSON_ID NUMBER generated as identity
    constraint PERSONS_PK
      primary key,
  NAME VARCHAR2(50),
  SURNAME VARCHAR2(50),
  PESEL VARCHAR2(11),
  CONTACT VARCHAR2(100)
)
```

1.2. Wycieczki

```
create table TRIPS
(
  TRIP_ID NUMBER generated as identity
    constraint TRIPS_PK
      primary key,
  NAME VARCHAR2(100),
  COUNTRY VARCHAR2(50),
  "DATE" DATE,
  DESCRIPTION VARCHAR2(200),
  SEATS_NUMBER NUMBER
)
```

1.3. Rezerwacje

```
create table RESERVATIONS
(
  RESERVATION_ID NUMBER generated as identity
    constraint RESERVATIONS_PK
      primary key,
  TRIP_ID NUMBER
    constraint RESERVATIONS_FK2
      references TRIPS,
  PERSON_ID NUMBER
    constraint RESERVATIONS_FK1
      references PERSONS,
  STATUS CHAR
    constraint RESERVATIONS_CHK1
      check (status IN ('N','P','Z','A'))
)
```

2. Przykładowe dane

2.1. Osoby

```
INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES('Archer', 'Moroney', '96514778264', 'tel: 979-224-677');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Paloma', 'Durrad', '63660990615', 'tel: 692-676-945');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Kimmi', 'Rangeley', '75840039207', 'tel: 571-261-518');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Charisse', 'Gavriel', '40311559253', 'tel: 374-572-464');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Jody', 'Frenchum', '55899624894', 'tel: 006-859-815');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Emmalynn', 'Sullly', '04982695028', 'tel: 126-696-852');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Angy', 'Murcutt', '95105220174', 'tel: 375-731-547');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Link', 'Fliege', '14690445316', 'tel: 147-961-102');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Adam', 'Kowalski', '87654321', 'tel: 662-322-331');

INSERT INTO PERSONS (NAME, SURNAME, PESEL, CONTACT)
VALUES ('Jan', 'Nowak', '12345678', 'tel: 231-439-564');
```

2.2. Wycieczki

```
INSERT INTO TRIPS (NAME, COUNTRY, "DATE", DESCRIPTION, SEATS_NUMBER)
VALUES ('Windsurfing na Prasonissi', 'Grecja', TO_DATE('2019-06-01
00:00:00', 'YYYY-MM-DD HH24:MI:SS'), 'Plaża i silne wiatry ...', 6);

INSERT INTO TRIPS (NAME, COUNTRY, "DATE", DESCRIPTION, SEATS_NUMBER)
VALUES ('Wycieczka do Paryza', 'Francja', TO_DATE('2016-01-01 00:00:00',
'YYYY-MM-DD HH24:MI:SS'), 'Ciekawa wycieczka ...', 3);

INSERT INTO TRIPS (NAME, COUNTRY, "DATE", DESCRIPTION, SEATS_NUMBER)
VALUES ('Piękny Kraków', 'Polska', TO_DATE('2017-02-03 00:00:00', 'YYYY-
MM-DD HH24:MI:SS'), 'Najciekawa wycieczka ...', 2);

INSERT INTO TRIPS (NAME, COUNTRY, "DATE", DESCRIPTION, SEATS_NUMBER)
VALUES ('Wieliczka', 'Polska', TO_DATE('2017-03-03 00:00:00', 'YYYY-MM-
DD HH24:MI:SS'), 'Zadziwiająca kopalnia ...', 2);
```

2.3. Rezerwacje

```
INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (4, 1, 'N');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (2, 2, 'P');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (1, 10, 'N');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (1, 8, 'P');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (1, 9, 'Z');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (1, 4, 'A');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (1, 3, 'Z');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (4, 7, 'A');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (4, 5, 'Z');

INSERT INTO RESERVATIONS (TRIP_ID, PERSON_ID, STATUS)
VALUES (2, 5, 'P');
```

3. Widoki

3.1. Wycieczki osoby

```
create or replace view V_TRIPS_PERSONS as
SELECT
  t.TRIP_ID,
  t.COUNTRY,
  t."DATE",
  t.NAME AS TRIP_NAME,
  p.PERSON_ID,
  p.NAME,
  p.SURNAME,
  r.STATUS AS RESERVATION_STATUS
FROM TRIPS t
  JOIN RESERVATIONS r ON t.TRIP_ID = r.TRIP_ID
  JOIN PERSONS p on r.PERSON_ID = p.PERSON_ID
```

3.2. Wycieczki osoby potwierdzone

```
create or replace view V_TRIPS_PERSONS_CONFIRMED as
SELECT TRIP_ID, COUNTRY, "DATE", TRIP_NAME, PERSON_ID, NAME, SURNAME,
RESERVATION_STATUS
FROM V_TRIPS_PERSONS
WHERE RESERVATION_STATUS = 'P' or RESERVATION_STATUS = 'Z'
```

3.3. Wycieczki przyszłe

```
create or replace view V_UPCOMING_TRIPS as
SELECT TRIP_ID, COUNTRY, "DATE", TRIP_NAME, PERSON_ID, NAME, SURNAME,
RESERVATION_STATUS
FROM V_TRIPS_PERSONS
WHERE "DATE" > CURRENT_DATE
```

3.4. Wycieczki miejsca

```
create or replace view V_TRIPS_SEATS as
SELECT
  t.TRIP_ID,
  t.COUNTRY,
  t."DATE",
  t.NAME AS TRIP_NAME,
  t.SEATS_NUMBER,
  t.SEATS_NUMBER - (SELECT COUNT(*) FROM RESERVATIONS r WHERE r.TRIP_ID
= t.TRIP_ID and r.STATUS != 'A') AS AVAILABLE_SEATS
FROM TRIPS t
```

3.5. Dostępne wycieczki

```
create or replace view V_AVAILABLE_TRIPS as
SELECT TRIP_ID, COUNTRY, "DATE", TRIP_NAME, SEATS_NUMBER,
AVAILABLE_SEATS
FROM V_TRIPS_SEATS
WHERE AVAILABLE_SEATS > 0 and "DATE" > CURRENT_DATE
```

3.6. Rezerwacje do anulowania

```
create or replace view V_RESERVATIONS_TO_CANCEL as
select
  r.RESERVATION_ID, t.NAME as TRIP_NAME, p.NAME, p.SURNAME, p.CONTACT
from RESERVATIONS r
  join PERSONS p on r.PERSON_ID = p.PERSON_ID
  join TRIPS t on r.TRIP_ID = t.TRIP_ID
where
  r.STATUS = 'N' and t."DATE" <= (CURRENT_DATE + 7)
```

4. Funkcje

4.1. Uczestnicy wycieczki

```
-- returned row type
create or replace type T_TP_RECORD as object (
  TRIP_ID number,
  COUNTRY varchar2(50),
  TRIP_DATE date,
  TRIP_NAME varchar2(100),
  PERSON_ID number,
  "NAME" VARCHAR2(50),
  SURNAME VARCHAR2(50),
  RESERVATION_STATUS CHAR
);

-- returned table type
create or replace type T_TP_TABLE as table of T_TP_RECORD;

-- function
create or replace function F_TRIP_PARTICIPANTS (v_trip_id in number)
return T_TP_TABLE as
  v_result T_TP_TABLE;
  v_id_count number;
begin

  -- check if trip exists
  select count(*) into v_id_count from TRIPS where TRIP_ID = v_trip_id;

  -- throw exception if not
  if v_id_count != 1 then
    raise_application_error(-20001, 'Trip with this id does not
exists');
  end if;

  -- select data
  select
    T_TP_RECORD(t.TRIP_ID, t.COUNTRY, t."DATE", t.TRIP_NAME,
t.PERSON_ID, t.NAME, t.SURNAME, t.RESERVATION_STATUS)
  bulk collect into
    v_result
  from V_TRIPS_PERSONS_CONFIRMED t
  where t.TRIP_ID = v_trip_id;

  -- return data
  return v_result;

end F_TRIP_PARTICIPANTS;
```

4.2. Rezerwacje osoby

```
-- function
create or replace function F_RESERVATIONS_PERSON (v_person_id in number)
return T_TP_TABLE as
  v_result T_TP_TABLE;
  v_id_count number;
begin

  -- check if person exists
  select count(*) into v_id_count from PERSONS where PERSON_ID =
v_person_id;

  -- throw exception if not
  if v_id_count != 1 then
    raise_application_error(-20002, 'Person with this id does not
exists');
  end if;

  -- select data
  select
    T_TP_RECORD(t.TRIP_ID, t.COUNTRY, t."DATE", t.TRIP_NAME,
t.PERSON_ID, t.NAME, t.SURNAME, t.RESERVATION_STATUS)
  bulk collect into
    v_result
  from V_TRIPS_PERSONS_CONFIRMED t
  where t.PERSON_ID = v_person_id;

  -- return data
  return v_result;

end F_RESERVATIONS_PERSON;
```

4.3. Przyszłe rezerwacje osoby

```
-- function
create or replace function F_UPCOMING_RESERVATIONS_PERSON (v_person_id
in number) return T_TP_TABLE as
  v_result T_TP_TABLE;
  v_id_count number;
begin

  -- check if person exists
  select count(*) into v_id_count from PERSONS where PERSON_ID =
v_person_id;

  -- throw exception if not
  if v_id_count != 1 then
    raise_application_error(-20002, 'Person with this id does not
exists');
  end if;

  -- select data
  select
    T_TP_RECORD(t.TRIP_ID, t.COUNTRY, t."DATE", t.TRIP_NAME,
t.PERSON_ID, t.NAME, t.SURNAME, t.RESERVATION_STATUS)
  bulk collect into
    v_result
  from V_UPCOMING_TRIPS t
  where t.PERSON_ID = v_person_id;

  -- return data
  return v_result;

end F_UPCOMING_RESERVATIONS_PERSON;
```

4.4. Dostępne wycieczki

```
-- returned row type
create or replace type T_AT_RECORD as object (
  TRIP_ID number,
  COUNTRY varchar2(50),
  TRIP_DATE date,
  TRIP_NAME varchar2(100),
  SEATS_NUMBER number,
  AVAILABLE_SEATS number
);

-- returned table type
create or replace type T_AT_TABLE as table of T_AT_RECORD;

-- function
create or replace function F_AVAILABLE_TRIPS (v_country in varchar2,
v_date_from in date, v_date_to in date) return T_AT_TABLE as
  v_result T_AT_TABLE;
begin

  -- throw dates are ic correct order
  if v_date_from > v_date_to then
    raise_application_error(-20003, 'Argument date_from has to be
earlier then date_to');
  end if;

  -- select data
  select
    T_AT_RECORD(t.TRIP_ID, t.COUNTRY, t."DATE", t.TRIP_NAME,
t.SEATS_NUMBER, t.AVAILABLE_SEATS)
  bulk collect into
    v_result
  from V_AVAILABLE_TRIPS t
  where t.COUNTRY = v_country and t."DATE" > v_date_from and t."DATE" <
v_date_to;

  -- return data
  return v_result;

end F_AVAILABLE_TRIPS;
```

5. Procedury

5.1. Dodaj rezerwacje

```
-- procedurre
create or replace procedure P_ADD_RESERVATION (v_trip_id in number,
v_person_id in number) as
  v_id_count number;
begin

  -- check if trip exists and is available
  select count(*) into v_id_count from V_AVAILABLE_TRIPS where TRIP_ID =
v_trip_id;

  -- throw exception if not
  if v_id_count != 1 then
    raise_application_error(-20004, 'Trip with this is unavailable or
does not exists');
  end if;

  -- check if person exists
  select count(*) into v_id_count from PERSONS where PERSON_ID =
v_person_id;

  -- throw exception if not
  if v_id_count != 1 then
    raise_application_error(-20002, 'Person with this id does not
exists');
  end if;

  -- check if there is no reservation for this person on this trip
  select count(*) into v_id_count from RESERVATIONS where TRIP_ID =
v_trip_id and PERSON_ID = v_person_id;

  -- throw exception if not
  if v_id_count != 0 then
    raise_application_error(-20005, 'This person already has reservation
for this trip');
  end if;

  -- create reservation
  insert into RESERVATIONS (TRIP_ID, PERSON_ID, STATUS) VALUES
(v_trip_id, v_person_id, 'N');


end P_ADD_RESERVATION;
```

5.2. Zmień status rezerwacji

```
-- procedurre
create or replace procedure P_CHANGE_RESERVATION_STATUS
(v_reservation_id in number, v_status in char) as
  v_id_count number;
begin

  -- check if reservation exists and don't have 'A' status
  select count(*) into v_id_count from RESERVATIONS where RESERVATION_ID
= v_reservation_id and STATUS != 'A';

  -- throw exception if not
  if v_id_count != 1 then
    raise_application_error(-20006, 'Reservation with this id does not
exists or has status A');
  end if;

  -- update reservation
  update RESERVATIONS SET STATUS = v_status WHERE RESERVATION_ID =
v_reservation_id;

end P_CHANGE_RESERVATION_STATUS;
```

5.3. Zmień liczbę miejsc

```
-- procedurre
create or replace procedure P_CHANGE_SEATS_NUMBER (v_trip_id in number,
v_seats_number in number) as
  v_id_count number;
begin

  -- check if trip exists
  select count(*) into v_id_count from TRIPS where TRIP_ID = v_trip_id;

  -- throw exception if not
  if v_id_count != 1 then
    raise_application_error(-20007, 'Trip with this id does not
exists');
  end if;

  -- check if change is possible
  select count(*) into v_id_count from V_TRIPS_SEATS where TRIP_ID =
v_trip_id and SEATS_NUMBER - AVAILABLE_SEATS <= v_seats_number;

  -- throw exception if not
  if v_id_count != 1 then
    raise_application_error(-20007, 'Cannot perform action because
reservations number exceeds new seats number');
  end if;

  -- update reservation
  update TRIPS SET SEATS_NUMBER = v_seats_number WHERE TRIP_ID =
v_trip_id;

end P_CHANGE_SEATS_NUMBER;
```

6. Dziennik statusu rezerwacji

6.1. Rezerwacje log

```
create table RESERVATIONS_LOG
(
  LOG_ID number generated as identity
      constraint RESERVATIONS_LOG_PK
        primary key,
  RESERVATION_ID number
      constraint RESERVATIONS_LOG_FK1
        references RESERVATIONS,
  LOG_DATE date
    default sysdate,
  STATUS char
)
```

6.2. [Zmiana] Dodawanie rezerwacji (procedura)

```
. . .

-- update reservation log
insert into RESERVATIONS_LOG (RESERVATION_ID, STATUS)
values (v_reservation_id, 'N');

. . .
```

6.3. [Zmiana] Zmiana statusu rezerwacji (procedura)

```
. . .

-- update reservation log
insert into RESERVATIONS_LOG (RESERVATION_ID, STATUS
values (v_reservation_id, v_status);

. . .
```

7. Dodanie pola ‘wolne miejsca’ do wycieczki

7.1. [Zmiana] Wycieczki (tabela)

```
alter table TRIPS
  add FREE_SEATS number;
```

7.2. Przeliczanie wolnych miejsc wycieczki

```
-- procedurre
create or replace procedure P_UPDATE_FREE_SEATS (v_trip_id in number) as
  v_taken_seats number;
begin

  -- get taken seats
  select count(*) into v_taken_seats from RESERVATIONS where TRIP_ID =
v_trip_id and STATUS != 'A';

  -- update trip
  update TRIPS set FREE_SEATS = SEATS_NUMBER - v_taken_seats where
TRIP_ID = v_trip_id;

end P_UPDATE_FREE_SEATS;
```

7.3. Przeliczanie wolnych miejsc wszystkich wycieczek

```
-- procedurre
create or replace procedure P_UPDATE_FREE_SEATS_ALL as
begin

  -- for each trip update its seats
  for v_trip in (select TRIP_ID FROM TRIPS) loop

    P_UPDATE_FREE_SEATS(V_TRIP_ID => v_trip.TRIP_ID);

  end loop;

end P_UPDATE_FREE_SEATS_ALL;
```

7.4. [Zmiana] Miejsca wycieczki (widok)

```
create or replace view V_TRIPS_SEATS_2 as
SELECT
  t.TRIP_ID,
  t.COUNTRY,
  t."DATE",
  t.NAME AS TRIP_NAME,
  t.SEATS_NUMBER,
  t.FREE_SEATS AS AVAILABLE_SEATS
FROM TRIPS t
```

7.5. [Zmiana] Dostępne wycieczki (widok)

```
create or replace view V_AVAILABLE_TRIPS_2 as
SELECT
  TRIP_ID, COUNTRY, "DATE", TRIP_NAME, SEATS_NUMBER, AVAILABLE_SEATS
FROM V_TRIPS_SEATS_2
WHERE AVAILABLE_SEATS > 0 and "DATE" > CURRENT_DATE
```

7.6. [Zmiana] Dostępne wycieczki (funkcja)

```
-- function
create or replace function F_AVAILABLE_TRIPS_2 (v_country in varchar2,
v_date_from in date, v_date_to in date) return T_AT_TABLE as
  v_result T_AT_TABLE;
begin

  -- throw if dates are in incorrect order
  if v_date_from > v_date_to then
    raise_application_error(-20003, 'Argument date_from has to be
earlier then date_to');
  end if;

  -- select data
  select
    T_AT_RECORD(t.TRIP_ID, t.COUNTRY, t."DATE", t.TRIP_NAME,
t.SEATS_NUMBER, t.AVAILABLE_SEATS)
  bulk collect into
    v_result
  from V_AVAILABLE_TRIPS_2 t
  where t.COUNTRY = v_country and t."DATE" > v_date_from and t."DATE" <
v_date_to;

  -- return data
  return v_result;

end F_AVAILABLE_TRIPS_2;
```

7.7. [Zmiana] Dodawanie rezerwacji (procedura)

```
. . .

-- update free seats
P_UPDATE_FREE_SEATS(V_TRIP_ID => v_trip_id);

. . .
```

7.8. [Zmiana] Zmiana statusu rezerwacji (procedura)

```
. . .

-- update free seats
select TRIP_ID into v_trip_id
from RESERVATIONS
where RESERVATION_ID = v_reservation_id;

P_UPDATE_FREE_SEATS(V_TRIP_ID => v_trip_id);

. . .
```

7.9. [Zmiana] Zmiana ilości miejsc wycieczki (procedura)

```
. . .

-- update free seats
P_UPDATE_FREE_SEATS(V_TRIP_ID => v_trip_id);

. . .
```

8. Wyzwalacze przy zmianie statusu

8.1. Zmiana lub utworzenie rezerwacji

```
create trigger TR_RESERVATION_CHANGE after insert or update on
RESERVATIONS
for each row
begin

  -- if there was a change in status
  if :NEW.STATUS != :OLD.STATUS then

    insert into RESERVATIONS_LOG (RESERVATION_ID, STATUS)
    values (:NEW.RESERVATION_ID, :NEW.STATUS);

  end if;

end;
```

8.2. Usunięcie rezerwacji

```
create trigger TR_RESERVATIONS_DELETE before delete on RESERVATIONS
begin
  raise_application_error(-20100, 'Reservations cannot be deleted');
end;
```

8.3. [Zmiana] Dodawanie rezerwacji (procedura)

```
. . .

-- update reservation log
insert into RESERVATIONS_LOG (RESERVATION_ID, STATUS)
values (v_reservation_id, 'N');

. . .
```

8.4. [Zmiana] Zmiana statusu rezerwacji (procedura)

```
. . .

-- update reservation log
insert into RESERVATIONS_LOG (RESERVATION_ID, STATUS
values (v_reservation_id, v_status);

. . .
```

9. Wyzwalacze przy zmianie ilości miejsc

9.1. Zmiana lub utworzenie wycieczki

```
create or replace trigger TR_TRIP_CHANGE before insert or update on
TRIPS
for each row
declare
  v_taken_seats number;
begin

  -- update free seats if there was a change in seats number
  if :NEW.SEATS_NUMBER != :OLD.SEATS_NUMBER then

    -- update has to be manual because it is impossible to call
P_UPDATE_FREE_SEATS

    -- get taken seats
    select count(*) into v_taken_seats from RESERVATIONS where TRIP_ID =
:NEW.TRIP_ID and STATUS != 'A';

    -- update trip
    :NEW.FREE_SEATS := :NEW.SEATS_NUMBER - v_taken_seats;

  end if;

end;
```

9.2. [Zmiana] Zmiana lub utworzenie rezerwacji (wyzwalacz)

```
. . .

-- update free seats
P_UPDATE_FREE_SEATS(:NEW.TRIP_ID);

-- update old trip if it was changed
if (:OLD.TRIP_ID is not null) and (:OLD.TRIP_ID != :NEW.TRIP_ID) then
  P_UPDATE_FREE_SEATS(:OLD.TRIP_ID);
end if;

. . .
```

9.3. [Zmiana] Zmiana ilości miejsc (procedura)

```
. . .

-- update free seats
P_UPDATE_FREE_SEATS(V_TRIP_ID => v_trip_id);

. . .
```

9.4. [Zmiana] Dodawanie rezerwacji (procedura)

```
. . .

-- update free seats
P_UPDATE_FREE_SEATS(V_TRIP_ID => v_trip_id);

. . .
```

9.5. [Zmiana] Zmiana statusu rezerwacji (procedura)

```
. . .

-- update free seats
select TRIP_ID into v_trip_id
from RESERVATIONS
where RESERVATION_ID = v_reservation_id;

P_UPDATE_FREE_SEATS(V_TRIP_ID => v_trip_id);

. . .
```