239537
Illia Azler

Lista 3 (TN Sr 7:30-9:00)

Zad1.
Baza:

```
create database firma;
use firma;
```

Tablicy:

```
use firma;
drop table if exists pracownicy;
drop table if exists ludzie;
drop table if exists zawody;

create table ludzie(
PESEL char(11) not null primary key,
imie varchar(30) not null,
nazwisko varchar(30) not null,
data_urodzenia date,
plec enum('K', 'M') not null);

create table zawody(
zawod_id int unsigned not null auto_increment primary key,
nazwa varchar(50) not null,
pensja_min float unsigned,
pensja_max float unsigned,
check (pensja_min<pensja_max));

create table pracownicy(
PESEL char(11) not null primary key,
zawod_id int unsigned not null,
pensja float unsigned);
```

Procedura wypewnienia danymi:

```
use firma;

insert into zawody(nazwa, pensja_min, pensja_max) values
        ('Polityk', 6000.00, 15000.00),
  ('Nauczyciel', 4500.00, 10000.00),
  ('Lekarz', 6000.00, 15000.00),
  ('Informatyk', 5500.00, 20000.00);
```

```
drop procedure if exists firma_ludzie_wypewnienie;
DELIMITER $$
create procedure firma_ludzie_wypewnienie(in count int, in new_pesel char(11), in data_u
date)
        begin
   declare x int;
   declare new_imie varchar(30);
   declare new_nazwisko varchar(30);
   declare wyplata float;
   declare new_plec varchar(1);
   declare pewno varchar(2);
   declare data_u_np varchar(20);

        set x = 0;

        loop_label:  loop
                if  x = count then
                        leave  loop_label;
                else
                        set data_u = DATE_ADD(data_u , INTERVAL 15 DAY);

        IF ((YEAR(CURRENT_DATE) - YEAR(data_u))-
                                (DATE_FORMAT(CURRENT_DATE, '%m%d') <
DATE_FORMAT(data_u, '%m%d')))<18 THEN
           set data_u_np = concat(substring(data_u,3,2), substring(data_u,6,2)+20,
substring(data_u,9,2));
           set pewno = 'NP';
                        ELSEIF ((YEAR(CURRENT_DATE) - YEAR(data_u))-
                                (DATE_FORMAT(CURRENT_DATE, '%m%d') <
DATE_FORMAT(data_u, '%m%d')))>60 THEN
           set data_u_np = concat(substring(data_u,3,2), substring(data_u,6,2),
substring(data_u,9,2));
           set pewno = 'P+';
                        ELSE
                                set data_u_np = concat(substring(data_u,3,2),
substring(data_u,6,2), substring(data_u,9,2));
           set pewno = 'P';
                        end if;


        set new_imie = concat('Imie_', pewno, '_', x+1);
                        set new_nazwisko = concat('Nazwisko_', pewno, '_', x+1);
                        set new_pesel = concat(substring(data_u,3,2), substring(data_u_np,3,2),
substring(data_u,9,2),
```

```
                        FLOOR(RAND()*(999-100+1))+100, FLOOR(RAND()*(9-1+1))+1,
FLOOR(RAND()*(9-1+1))+1);
                IF (substring(new_pesel,10,1) = "2"
                        or substring(new_pesel,10,1) = "4"
                        or substring(new_pesel,10,1) = "6"
                        or substring(new_pesel,10,1) = "8") THEN
                                set new_plec = "K";
                ELSEIF (substring(new_pesel,10,1) = "1"
                        or substring(new_pesel,10,1) = "3"
                        or substring(new_pesel,10,1) = "5"
                        or substring(new_pesel,10,1) = "7"
                        or substring(new_pesel,10,1) = "9") THEN
                                set new_plec = "M";
                end if;

        insert into ludzie (PESEL, imie, nazwisko, data_urodzenia, plec) values
                        (new_pesel, new_imie, new_nazwisko, data_u, new_plec);
        set x = x + 1;
                end if;
         end loop;
         end$$
DELIMITER ;
call firma_ludzie_wypewnienie(5, "04210131234", "2004-01-01");
call firma_ludzie_wypewnienie(45, "95120113245", "1995-12-01");
call firma_ludzie_wypewnienie(5, "45020112311", "1945-02-01");
```

Triggery dla prawidłowego formatu PESEL:

```
use firma;
drop trigger if exists pesel_ludzie_insert_check;
drop trigger if exists pesel_ludzie_update_check;
drop trigger if exists pesel_pracownicy_insert_check;
drop trigger if exists pesel_pracownicy_update_check;

DELIMITER $$
create trigger pesel_ludzie_insert_check before insert on ludzie for each row
begin
        DECLARE r_pesel varchar(2);
        DECLARE m_pesel varchar(2);
        DECLARE d_pesel varchar(2);
        DECLARE p_pesel varchar(1);
   declare data_u date;

   SET r_pesel = substring(NEW.PESEL,1,2);
   SET m_pesel = substring(NEW.PESEL,3,2);
```

```
  SET d_pesel = substring(NEW.PESEL,5,2);
  SET p_pesel = substring(NEW.PESEL,10,1);
  set data_u = new.data_urodzenia;

  IF (LENGTH(NEW.PESEL)) != 11 THEN
                        signal sqlstate "11111"
                        set message_text = "PESEL ma 11 symbolow";
  ELSEIF (substring(NEW.data_urodzenia,3,2) != r_pesel) THEN
                                signal sqlstate "11111"
                                set message_text = "PESEL sie nie zgadza z data(rok)";
        IF ((YEAR(CURRENT_DATE) - YEAR(data_u))-
                                (DATE_FORMAT(CURRENT_DATE, '%m%d') <
DATE_FORMAT(data_u, '%m%d')))<18 THEN
                IF (substring(NEW.data_urodzenia,6,2) != m_pesel+20) THEN
                        signal sqlstate "11111"
                        set message_text = "PESEL sie nie zgadza z data(miesac)";
                end if;
        ELSE
                IF (substring(NEW.data_urodzenia,6,2) != m_pesel+20) THEN
                                signal sqlstate "11111"
                                set message_text = "PESEL sie nie zgadza z data(miesac)";
                end if;
        end if;

        ELSEIF (substring(NEW.data_urodzenia,9,2) != d_pesel) THEN
                                signal sqlstate "11111"
                                set message_text = "PESEL sie nie zgadza z data(dzien)";

        ELSEIF (NEW.plec = "K"
                and p_pesel not like "2"
                and p_pesel not like "4"
                and p_pesel not like "6"
                and p_pesel not like "8") THEN
                        signal sqlstate "11111"
                        set message_text = "Niepoprawny PESEL dla K";
  ELSEIF (NEW.plec = "M"
                and p_pesel not like "1"
                and p_pesel not like "3"
                and p_pesel not like "5"
                and p_pesel not like "7"
                and p_pesel not like "9") THEN
                        signal sqlstate "11111"
                        set message_text = "Niepoprawny PESEL dla M";
  end if;
end$$
```

```
create trigger pesel_ludzie_update_check before update on ludzie for each row
begin
   IF (LENGTH(NEW.PESEL))!=11 THEN
                        signal sqlstate '11111'
                        set message_text = 'Niedozwolony format PESEL';
   end if;
end$$

DELIMITER ;
```

Kursor:

```
drop procedure if exists firma_pracownicy_wypewnienie;

DELIMITER $$

CREATE PROCEDURE firma_pracownicy_wypewnienie()

BEGIN
   DECLARE done INT DEFAULT FALSE;
   DECLARE pes char(11);
   DECLARE d_u date;
   DECLARE state enum('K', 'M');
   declare zawod int unsigned;
   declare wyplata float unsigned;

        DECLARE mycursor CURSOR FOR(
                select PESEL, data_urodzenia, plec from ludzie
                where (YEAR(CURRENT_DATE) - YEAR(data_urodzenia))>18);
        DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

   OPEN mycursor;

   fetch_loop: LOOP
   FETCH mycursor INTO pes, d_u, state;
        IF done THEN
     LEAVE fetch_loop;
   END IF;

   select zawody.zawod_id into zawod from zawody order by rand() limit 1;
   select RAND()*(zawody.pensja_max-zawody.pensja_min)+zawody.pensja_min into wyplata
from zawody where zawod_id like zawod;
   if (zawod like '3'
                and ((YEAR(CURRENT_DATE) - YEAR(d_u))-
                                (DATE_FORMAT(CURRENT_DATE, '%m%d') < DATE_FORMAT(d_u,
'%m%d')))>65
```

```
                and state like 'M') then
                        select pes;
        elseif (zawod like '3'
                and ((YEAR(CURRENT_DATE) - YEAR(d_u))-
                                (DATE_FORMAT(CURRENT_DATE, '%m%d') < DATE_FORMAT(d_u,
'%m%d')))>60
                and state like 'K') then
                        select pes;
   else
                insert into pracownicy(PESEL, zawod_id, pensja) values
                (pes, zawod, wyplata);
        end if;
 END LOOP;
 CLOSE mycursor;
END$$
DELIMITER ;

call firma_pracownicy_wypewnienie;
```

Wniosek:
PESEL, moim zdaniem, musi zostać primary key, ponieważ PESEL jest unikalny. Wszędzie zachowano polecenia.

Zad2.

```
-- CREATE INDEX plec_imie ON ludzie(plec, imie);
-- CREATE INDEX pensja ON pracownicy(pensja);

-- DROP INDEX plec_imie ON ludzie;
-- DROP INDEX pensja ON ludzie;

show index from firma.ludzie;
show index from firma.pracownicy;

explain select imie, plec from ludzie where plec like 'K';
-- and imie like "A%"

explain select plec from ludzie where plec like 'K';
explain select imie from ludzie where imie like 'K%';
explain select * from pracownicy where pensja < 2000;
explain select ludzie.PESEL, pracownicy.pensja from (ludzie
join pracownicy on pracownicy.pesel = ludzie.pesel)
where pracownicy.pensja>10000 and ludzie.plec like 'M' and pracownicy.zawod_id like '4';
```

Te które utworzyliśmy i do tego jeszcze primary key w każdej tabeli.
Indeks służy do szybkiego obliczenia zapytań na indeksowanych kolumnach, bardziej widoczne na wielkich bazach danych.

Zad3.

```
use firma;
drop procedure if exists firma_pensja_podwyzka;

DELIMITER $$
create procedure firma_pensja_podwyzka(in def_nazwa varchar(30))
BEGIN
        declare ilosc_ogolna int;
   declare ilosc_war int;

        select count(*) into ilosc_ogolna from (zawody
        join pracownicy on pracownicy.zawod_id = zawody.zawod_id)
        where zawody.nazwa = def_nazwa;

   select count(*) into ilosc_war from (zawody
        join pracownicy on pracownicy.zawod_id = zawody.zawod_id)
        where zawody.nazwa like def_nazwa
                and (105*pracownicy.pensja/100) > zawody.pensja_min
                and (105*pracownicy.pensja/100) < zawody.pensja_max;

   set autocommit = 0;
        START TRANSACTION;
                UPDATE firma.pracownicy SET pensja = (105*pracownicy.pensja/100)
      WHERE pracownicy.zawod_id = (select zawod_id from zawody where nazwa = def_nazwa
having ilosc_ogolna = ilosc_war) limit 100;
        COMMIT;
END$$;
DELIMITER ;

call firma_pensja_podwyzka("Informatyk");

/*
DELIMITER $$
create procedure firma_pensja_podwyzka(in nazwa_z varchar(30))
        BEGIN
   DECLARE done INT DEFAULT FALSE;
   DECLARE pes char(11);
   declare id int unsigned;
   declare wyplata float unsigned;
   declare wyplata_min float unsigned;
   declare wyplata_max float unsigned;


        DECLARE mycursor CURSOR FOR(
```

```
                select pracownicy.PESEL, pracownicy.zawod_id, pracownicy.pensja,
zawody.pensja_min, zawody.pensja_max from (zawody
                join pracownicy on pracownicy.zawod_id = zawody.zawod_id)
                where pracownicy.zawod_id like id
      order by pracownicy.pensja desc);
        DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

  select zawod_id into id from zawody where nazwa_z like zawody.nazwa;

  OPEN mycursor;

  fetch_loop: LOOP
  FETCH mycursor INTO pes, id, wyplata, wyplata_min, wyplata_max;
        IF done THEN
   LEAVE fetch_loop;
  END IF;
  if ((105*wyplata/100)>wyplata_min
                and (105*wyplata/100)<wyplata_max)then
                        UPDATE firma.pracownicy SET pensja = (105*wyplata/100)  WHERE
(PESEL = pes);
        else
                signal sqlstate "11111"
                set message_text = "Nikt nie dostanie podwyzki, jeden ma wiecej od max pensji";
  end if;
 END LOOP;
 CLOSE mycursor;
END$$
DELIMITER ;
call firma_pensja_podwyzka("Informatyk");
*/
```

Mam dwie wesji, ponieważ nie zauważyłem, że musi być tranzakcja, obydwa działają pod warunkiem.

Zad4.

```
SET @sql = "select count(plec) from ((ludzie
join pracownicy on pracownicy.PESEL = ludzie.PESEL)
join zawody on zawody.zawod_id = pracownicy.zawod_id)
where ludzie.plec like 'K' and zawody.nazwa = (?)";
PREPARE getIloscKobietSql FROM @sql;
SET @zawod = 'Informatyk';
EXECUTE getIloscKobietSql using @zawod;
```

Zad5.

Nie mam polecen, korzystam z GUI, MySQL Workbench, jednak proces backup'u jest podobny.

Pewny backup różni się od różnicowego tym, że różnicowy backup nie całej bazy, a tylko części, gdzie były zmiany.

Zad6.
Niestety nie udało się realizować przez brak czasu ;(