

KAUNO TECHNOLOGIJOS UNIVERSITETAS

INFORMATIKOS FAKULTETAS

INFORMACINIŲ SISTEMŲ STUDIJŲ PROGRAMA

Marius Krajauskas

**Duomenų bazių laboratorinių darbų ataskaita**

Vadovas

ALGIRDAS ŠUKYS

KAUNAS, 2014

Turinys

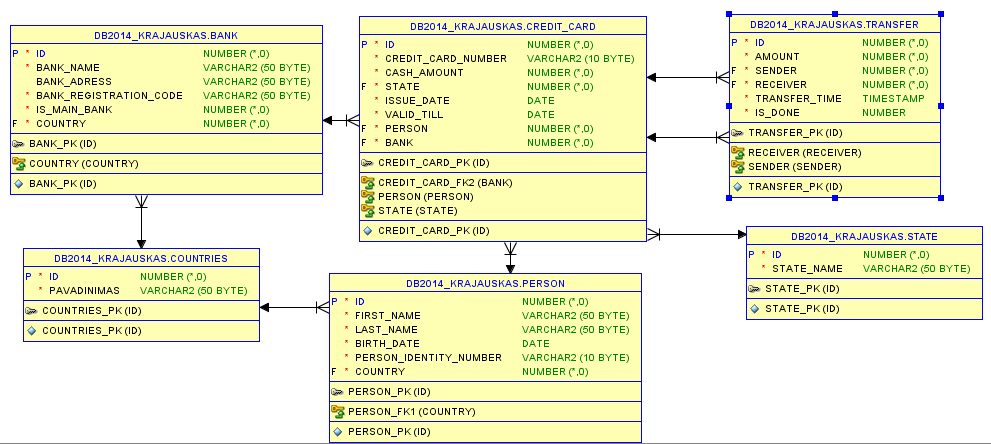
[1. Duomenų bazės projektavimas 3](#_Toc399604094)

[2. SQL Užklausos 4](#_Toc399604095)

# Duomenų bazės projektavimas

Aš šiam laboratoriniam darbui sukūriau 5 lenteles bankų sistemos tematika. Jose bus saugoma šalys, bankai, vartotojai, jų kreditinės kortelės, bei atliekami pinigų pavedimai.

Suprojektuotos duomenų bazės schema:



# SQL Užklausos

Duomenų bazę užpildau duomenimis šiomis INSERT užklausomis:

## Insert

**Šalių įvedimas:**

insert into COUNTRIES (PAVADINIMAS) values ('Lietuva');

insert into COUNTRIES (PAVADINIMAS) values ('Anglija');

insert into COUNTRIES (PAVADINIMAS) values ('Latvija');

insert into COUNTRIES (PAVADINIMAS) values ('Estija');

insert into COUNTRIES (PAVADINIMAS) values ('Lenkija');

insert into COUNTRIES (PAVADINIMAS) values ('Suomija');

insert into COUNTRIES (PAVADINIMAS) values ('Vokietija');

insert into COUNTRIES (PAVADINIMAS) values ('Amerika');

insert into COUNTRIES (PAVADINIMAS) values ('Rusija');

insert into COUNTRIES (PAVADINIMAS) values ('Švedija');

**Bankų įvedimas:**

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('Lietuvos bankas','liepu gatve 5','145645',1 ,1);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('Siauliu bankas','azuolu gatve 15','654214',0 ,1);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('SWED','svedijos gatve 72','544725',0 ,4);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('BANK2','gatve 1','000001',0 ,4);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('BANK3','gatve 2','000002',1 ,4);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('BANK4','gatve 3','000003',0 ,5);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('BANK5','gatve 4','000004',1 ,5);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('BANK6','gatve 5','000005',0 ,6);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('BANK7','gatve 6','000006',0 ,8);

insert into BANK (BANK\_NAME, BANK\_ADRESS, BANK\_REGISTRATION\_CODE, IS\_MAIN\_BANK, COUNTRY ) values ('BANK8','gatve 7','000007',1 ,10);

**Banko kortelės naudotojų įvedimas:**

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('Jonas','Jonaitis', TO\_DATE('1992/09/01', 'yyyy/mm/dd'),'10000000',1);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS','PAVARDE', TO\_DATE('1992/01/09', 'yyyy/mm/dd'),'10000002',1);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS2','PAVARDE2', TO\_DATE('1991/02/08', 'yyyy/mm/dd'),'10000003',4);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS3','PAVARDE3', TO\_DATE('1990/03/07', 'yyyy/mm/dd'),'10000004',2);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS4','PAVARDE4', TO\_DATE('1989/04/06', 'yyyy/mm/dd'),'10000005',4);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS5','PAVARDE5', TO\_DATE('1988/05/05', 'yyyy/mm/dd'),'10000006',5);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS6','PAVARDE6', TO\_DATE('1987/06/04', 'yyyy/mm/dd'),'10000007',8);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS7','PAVARDE7', TO\_DATE('1986/07/03', 'yyyy/mm/dd'),'10000008',7);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS8','PAVARDE8', TO\_DATE('1985/08/02', 'yyyy/mm/dd'),'10000009',8);

insert into PERSON (FIRST\_NAME, LAST\_NAME, BIRTH\_DATE,PERSON\_IDENTITY\_NUMBER, COUNTRY)

values ('VARDAS9','PAVARDE9', TO\_DATE('1980/10/01', 'yyyy/mm/dd'),'10000010',10);

**Bankų kortelių būsenų įvedimas:**

insert into STATE (STATE\_NAME) values ('Registruota');

insert into STATE (STATE\_NAME) values ('Aktyvi');

insert into STATE (STATE\_NAME) values ('Blokuota');

insert into STATE (STATE\_NAME) values ('Išėjus iš galiojimo');

**Bankų kortelių įvedimas:**

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000001', 500, 1, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 1, 1 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000002', 200, 2, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 2, 1 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000003', 100, 2, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 3, 1 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000004', 50, 2, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 4, 2 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000005', 5000, 3, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 5, 4 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000006', 35000, 2, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 6, 6 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000007', 506, 2, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 7, 5 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000008', 3254, 2, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 8, 9 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000009', 1541, 3, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 9, 10 );

INSERT INTO CREDIT\_CARD (CREDIT\_CARD\_NUMBER, CASH\_AMOUNT, STATE, ISSUE\_DATE, VALID\_TILL, PERSON, BANK)

VALUES ( '0000000010', 120, 2, TO\_DATE('2014/09/27', 'yyyy/mm/dd'), TO\_DATE('2014/10/01', 'yyyy/mm/dd'), 10, 5 );

**Banko pavedimų įvedimas:**

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (200, 1, 2, timestamp '2014-09-09 12:23:45',0);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (100, 2, 3, timestamp '2014-09-09 12:23:45',0);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (500, 3, 1, timestamp '2014-09-09 12:23:45',0);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (50, 4, 5, timestamp '2014-09-09 12:23:45',0);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (400, 5, 6, timestamp '2014-09-09 12:23:45',1);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (2000, 6, 4, timestamp '2014-09-09 12:23:45',0);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (40, 7, 8, timestamp '2014-09-09 12:23:45',0);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (260, 8, 9, timestamp '2014-09-09 12:23:45',1);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (150, 9, 7, timestamp '2014-09-09 12:23:45',1);

insert into TRANSFER (AMOUNT, SENDER, RECEIVER, TRANSFER\_TIME, IS\_DONE) values (400, 6, 7, timestamp '2014-09-09 12:23:45',0);

## UPDATE

update PERSON set FIRST\_NAME='Vardenis', LAST\_NAME='Pavardenis' where ID=10;

## DELETE

delete person where ID=11;

## SELECT

Su šia užklausa pasirenku visų bankų kreditų kortelių pinigų sumą taip pat miestus, kuriuose tie bankai yra.

WITH SUM\_TABLE AS(

select CREDIT\_CARD.BANK as BANK\_ID, SUM(CREDIT\_CARD.CASH\_AMOUNT) AS SUMA from CREDIT\_CARD GROUP BY CREDIT\_CARD.BANK

)

SELECT BANK.BANK\_NAME, BANK.BANK\_ADRESS, COUNTRIES.PAVADINIMAS, BANK.BANK\_REGISTRATION\_CODE, SUMA

FROM COUNTRIES inner join BANK on COUNTRIES.ID = BANK.COUNTRY inner join SUM\_TABLE on BANK.ID=BANK\_ID;

# Vaizdiniai. View

Šiame laboratoriniame darbe sukūriau View lentelę, kuri išveda anksčiau aprašytos SELECT užklausos rezultatus su sąlyga jei tų bankų turima pinigų suma yra didesnė už 1000.

Užklausa:

WITH SUM\_TABLE AS(

select CREDIT\_CARD.BANK as BANK\_ID, SUM(CREDIT\_CARD.CASH\_AMOUNT) AS SUMA from CREDIT\_CARD GROUP BY CREDIT\_CARD.BANK having SUM(CREDIT\_CARD.CASH\_AMOUNT)>1000

)

SELECT BANK.BANK\_NAME, BANK.BANK\_ADRESS, COUNTRIES.PAVADINIMAS, BANK.BANK\_REGISTRATION\_CODE, SUMA

FROM COUNTRIES inner join BANK on COUNTRIES.ID = BANK.COUNTRY inner join SUM\_TABLE on BANK.ID=BANK\_ID

# Funkcijos ir procedūros.

Šioje laboratorinio darbo dalyje sukūriau funkciją, į kurią padavus banko id ji suranda to banko turimą pinigų sumą.

CREATE OR REPLACE FUNCTION SUM\_CASH\_BY\_BANK

( BANK\_ID IN NUMBER

) RETURN NUMBER AS SUMA CREDIT\_CARD.CASH\_AMOUNT%TYPE;

BEGIN

select SUM(CREDIT\_CARD.CASH\_AMOUNT) into SUMA from CREDIT\_CARD where CREDIT\_CARD.BANK = BANK\_ID GROUP BY CREDIT\_CARD.BANK;

IF SUMA > 0

then RETURN SUMA;

else

SUMA := cast(0 as number);

return SUMA;

END IF;

END SUM\_CASH\_BY\_BANK;

Sukurta procedūrą atlikti pavedimui iš vienos sąskaitos į kitą.

CREATE OR REPLACE PROCEDURE MONEY\_TRANSFER

(

SENDER IN NUMBER

, RECEIVER IN NUMBER

, AMOUNT IN NUMBER

) AS

BEGIN

update CREDIT\_CARD set CASH\_AMOUNT = CASH\_AMOUNT - AMOUNT WHERE CREDIT\_CARD.ID = SENDER;

update CREDIT\_CARD set CASH\_AMOUNT = CASH\_AMOUNT + AMOUNT WHERE CREDIT\_CARD.ID = RECEIVER;

END MONEY\_TRANSFER;

# Trigeriai, Indeksai

Šiame laboratoriniame darbe sukūriau trigerį, kuris pinigų pavedimo sukūrimo metu perveda nurodytą sumą iš vienos sąskaitos į kitą. Šis trigeris iškviečia anksčiau sukurtą procedūrą, kuri įvykdo pavedimą.

CREATE OR REPLACE TRIGGER TRANSACTION\_TRIGGER\_NEW

AFTER INSERT OR UPDATE OF IS\_DONE ON TRANSFER

FOR EACH ROW

BEGIN

MONEY\_TRANSFER(:NEW.SENDER, :NEW.RECEIVER, :NEW.AMOUNT);

END;

Indeksai

Sukurtas indeksas, kuris palengvina paieška pagal kreditinės kortelės turimą pinigų sumą ir išdavimo data. Užklausa parodanti indekso veikimą.

SELECT \* FROM CREDIT\_CARD order by CASH\_AMOUNT desc, ISSUE\_DATE;