

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

mācību priekšmetā

“Lielu datu bāzu tehnoloģijas (DB2)”

4.praktiskais darbs

‘Programmēšanas valodu PL/SQL un Java izmantošana relāciju-objektu datu bāzē’

Izstrādāja: Jolanta Bērziņa

Stud.apl.nr 181RDB178

2020./2021. m. g.

**Saturs**

[Ievads 3](#_Toc60773891)

[1. Datu bāzes shēmas 4](#_Toc60773892)

[1.1. Datu bāzes ierakstu shēma 4](#_Toc60773893)

[1.2. Objektu hierarhijas shēma 5](#_Toc60773894)

[1.3. Skatu shēmas 6](#_Toc60773895)

[2. Datubāzes izveide 7](#_Toc60773896)

[2.1. Tabulu un metožu definēšana 7](#_Toc60773897)

[2.1.1. Tabulas NOMAS\_FIRMA un iesaistīto objektu tipu izveide 7](#_Toc60773898)

[2.1.2. Skata SKATS\_NOMAS\_AUTO izveide 9](#_Toc60773899)

[2.1.3. Tabulas REKINI un iesaistīto objektu tipu izveide 10](#_Toc60773900)

[2.1.4. Tabulas KLIENTI un iesaistīto objektu tipu izveide 10](#_Toc60773901)

[2.1.5. Skata SKATS\_KLIENTI izveide 12](#_Toc60773902)

[2.2. Datu ievade 14](#_Toc60773903)

[2.2.1. Tabula NOMAS\_FIRMAS 14](#_Toc60773904)

[2.2.2. Tabula KLIENTI 16](#_Toc60773905)

[3. VAICĀJUMI UN METOŽU IZMANTOŠANA 19](#_Toc60773906)

[3.1. Metode IZVADE 19](#_Toc60773907)

[3.2. Metode PARBAUDE 19](#_Toc60773908)

[3.3. Metode SALIDZINAT\_NOMAS\_CENU 21](#_Toc60773909)

[3.4. MAP Metode BRAUKSANAS\_STAZS 21](#_Toc60773910)

[3.5. Member metode BRAUKSANAS\_PIEREDZE java 22](#_Toc60773911)

[3.6. Metode CIK\_JAMAKSA 22](#_Toc60773912)

[Secinājumi 23](#_Toc60773913)

# Ievads

Ceturtā praktiskā darba ietvaros ir jāatspoguļo līdz šiem apgūtās zināšanas saistībā ar relāciju-objektu tabulām, tipiem, metodēm un skatiem. Papildus ir jārealizē mantošanas un pārrakstīšanas iespējas un jāizveido metodes PL/SQL un Java valodās.

Ceturtajā praktiskajā darbā ir jārealizē sekojoši uzdevumi:

1. Jāizveido no divām vai vairākām saliktām objektu rindas tabulām. Tabulās ir:
   1. Pakārtotie objektu;
   2. Objektu kolekcijas;
   3. Objektu hierarhijas (heterogēnie objekti).
2. Tabulas objektiem ir jāizveido metodes PL/SQL un Java valodās. Objektu tipu hierarhijai jāparāda arī mantošanas un pārrakstīšanas iespējas.
3. Jāizveido objektu skati:
   1. Lai sasaistītu objektu tabulas ar objektu saitēm (REF(), DEREF() izmantošana);
   2. Lai sasaistītu datus objekta tabulā (REF(), DEREF() izmantošana);
   3. Lai izveidotu vajadzīgās metodes ar objektu skatu palīdzību.
4. Divām objektu metodēm jābūt realizētām Java valodā.
5. Obligāti jāuzzīmē datu bāzes shēma:
   1. Datiem;
   2. Saitēm (OID, REF(), DEREF());
   3. Metodēm;
6. Jāveic vaicājumi, lai nodemonstrētu, ka viss funkcionē.
7. Jāveic secinājumi, un ieteikumi apskatītai tehnoloģijām un lekcijām.

# Datu bāzes shēmas

## Datu bāzes ierakstu shēma

F\_NUM

F\_NOS

OID

KRAVAS AUTO

KRAVAS AUTO

VALSTS

PILSETA

IELA

P-INDEKSS

TIPS\_ADRESE

VIEGLAIS AUTO

VIEGLAIS AUTO

NOMAS\_AUTO

AUTOBUSS

TIPS\_NOMASFIRMA

**Metode:** CONSTRUCTOR

**Metode:** IZVADE

K\_NUM

K\_UZV

K\_VAR

PERS\_KODS

OID

VAD\_APLIECIBA

NOMAS\_VESTURE

REF\_REKINS

REF\_REKINS

REF\_REKINS

BANKA

KONTS

TIPS\_NOREKINS

BONUSA\_LIMENIS

TIPS\_KLIENTS

**Metode:** CONSTRUCTOR

**Metode:** BRAUKSANAS \_PIEREDZE

R\_NUM

LAIKS

CENA

A\_NUM

SUMMA

F\_NUM

TIPS\_REKINS

K\_NUM

REF\_AUTO

**Metode:** CONSTRUCTOR

TABULA

NOMAS\_FIRMAS

TABULA

KLIENTI

TABULA

REKINI

1:N

## Objektu hierarhijas shēma

**TIPS\_AUTO**

NUM

MARKA

MODELIS

GADS

NOBRAUKUMS

CENA

NOMU SKAITS

**KRAVAS\_AUTO**

KRAVAS\_IETILPIBA

**AUTOBUSS**

PASAZIERU\_SK

**Metode:** PARBAUDE

**Metode:** PARBAUDE

**Metode:** PARBAUDE

## Skatu shēmas

TABULA **NOMAS\_FIRMAS**

TIPS**\_NOMAS\_AUTO**

**SKATS\_NOMAS\_AUTO**

**Metode:** SALIDZINAT \_NOMAS\_CENU

TABULA **REKINI**

TIPS**\_SKATS\_REKINI**

**TABULA KLIENTI**

**TIPS\_SKATS\_KLIENTI**

**SKATS\_REKINI**

**Metode:** gala\_summa

**Metode:** BRAUKSANAS\_STAZS

# Datubāzes izveide

## Tabulu un metožu definēšana

### Tabulas NOMAS\_FIRMA un iesaistīto objektu tipu izveide

Objektu tipa T\_ADRESE izveide:

create or replace type T\_ADRESE as object(

VALSTS varchar2(20),

PILSETA varchar2(30),

IELA varchar2(25),

P\_INDEKSS varchar2(10)

);

Objekta tipa T\_AUTO izveide:

create or replace type T\_AUTO as object (

A\_NUM number,

MARKA varchar2(20),

MODELIS varchar2(20),

GADS number,

NOBRAUKUMS number,

CENA number,

NOMU\_SK number,

SKATE date,

member function PARBAUDE return number

) not final;

create or replace TYPE BODY T\_AUTO AS

member function PARBAUDE return number AS

BEGIN

RETURN trunc(self.skate - sysdate );

END PARBAUDE;

END;

Apakštipa KRAVAS\_AUTO izveide:

create or replace type KRAVAS\_AUTO under T\_AUTO (

K\_IETILPIBA number,

overriding member function PARBAUDE return number

);

create or replace TYPE BODY KRAVAS\_AUTO AS

overriding member function PARBAUDE return number AS

BEGIN

RETURN trunc (self.skate - sysdate-10);

END PARBAUDE;

End;

Apakštipa AUTOBUSS izveide:

create or replace type AUTOBUSS under T\_AUTO (

PASAZIERU\_SK number,

overriding member function PARBAUDE return number

);

create or replace type body AUTOBUSS as

overriding member function PARBAUDE return number is

begin

if self.SKATE-sysdate >30

then return trunc(self.SKATE-sysdate - 30);

else return trunc(self.SKATE-sysdate);

end if;

end PARBAUDE ;

end;

create or replace type KOLEKCIJA\_AUTO as table of T\_AUTO;

Objekta tipa TIPS\_AUTONOMAS\_FIRMA izveide ar *constructor* metodi:

create or replace type TIPS\_AUTONOMAS\_FIRMA as object (

F\_NUM number,

F\_NOS varchar2(25),

F\_ADRESE T\_ADRESE,

NOMAS\_AUTO KOLEKCIJA\_AUTO,

constructor function TIPS\_AUTONOMAS\_FIRMA (F\_NUM number, F\_NOS varchar2, F\_ADRESE T\_ADRESE) return SELF as result,

member procedure IZVADE

);

create or replace TYPE BODY TIPS\_AUTONOMAS\_FIRMA AS

constructor function TIPS\_AUTONOMAS\_FIRMA (F\_NUM number, F\_NOS varchar2, F\_ADRESE T\_ADRESE) return SELF as result AS

BEGIN

SELF.F\_NUM:=F\_NUM;

SELF.F\_NOS:=F\_NOS;

SELF.F\_ADRESE:=F\_ADRESE;

RETURN ;

END TIPS\_AUTONOMAS\_FIRMA;

member procedure IZVADE is

begin

dbms\_output.put\_line('Firmas numurs: '|| F\_NUM);

dbms\_output.put\_line('Firmas nosaukums: '|| F\_NOS);

dbms\_output.put\_line('Firmas adrese: '|| F\_ADRESE.VALSTS || F\_ADRESE.PILSETA || F\_ADRESE.IELA|| F\_ADRESE.P\_INDEKSS);

end IZVADE;

END;

create table NOMAS\_FIRMAS of TIPS\_AUTONOMAS\_FIRMA

nested table NOMAS\_AUTO store as TAB\_NOMAS\_AUTO;

### Skata SKATS\_NOMAS\_AUTO izveide

Skata SKATS\_NOMAS\_AUTO un tā atbilstošā objekta tipa TIPS\_NOMAS\_AUTO izveide:

create or replace type TIPS\_NOMAS\_AUTO as object (

F\_NUM number,

A\_NUM number,

MARKA varchar2(20),

MODELIS varchar2(20),

GADS number,

NOBRAUKUMS number,

CENA number,

NOMU\_SK number,

SKATE date,

K\_IETILPIBA number,

PASAZIERU\_SK number,

ORDER member function SALIDZINAT\_NOMAS\_CENU(m\_obj TIPS\_NOMAS\_AUTO) return integer

);

CREATE OR REPLACE

TYPE BODY TIPS\_NOMAS\_AUTO AS

ORDER member function SALIDZINAT\_NOMAS\_CENU(m\_obj TIPS\_NOMAS\_AUTO) return integer IS

BEGIN

If SELF.CENA < m\_obj.CENA then return -1;

elsif SELF.CENA > m\_obj.CENA then return 1;

else return 0;

end if;

RETURN NULL;

END SALIDZINAT\_NOMAS\_CENU;

END;

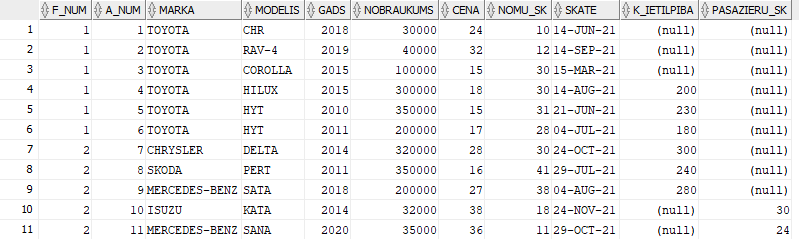
create or replace view SKATS\_NOMAS\_AUTO of TIPS\_NOMAS\_AUTO

with object identifier(A\_NUM) AS

select A.F\_NUM, VALUE(B).A\_NUM, VALUE(B).MARKA, VALUE(B).MODELIS, VALUE(B).GADS, VALUE(B).NOBRAUKUMS, VALUE(B).CENA, VALUE(B).NOMU\_SK,

VALUE(B).SKATE, TREAT(VALUE(B) AS KRAVAS\_AUTO).K\_IETILPIBA, TREAT(VALUE(B) AS AUTOBUSS).PASAZIERU\_SK

from NOMAS\_FIRMAS a, TABLE (value(a).NOMAS\_AUTO) B;



2.1.2.1.attēls. Skata SKATS\_NOMAS\_AUTO ieraksti

### Tabulas REKINI un iesaistīto objektu tipu izveide

Objekta tipa TIPS\_REKINS izveide, kas izmanto atsauci uz iepriekš izveidoto skatu SKATS\_NOMAS\_AUTO:

create or replace type T\_REKINS as object(

R\_NUM number,

K\_NUM number,

F\_NUM number,

A\_NUM number,

CENA number,

LAIKS number,

REKINA\_SUMMA number,

REF\_AUTO ref TIPS\_NOMAS\_AUTO,

constructor function T\_REKINS (R\_NUM number, K\_NUM number, F\_NUM number, A\_NUM number, LAIKS number) return self as result

create or replace type BODY T\_REKINS as

constructor function T\_REKINS (R\_NUM number, K\_NUM number, F\_NUM number, A\_NUM number, LAIKS number)

return self as result is

BEGIN

SELF.R\_NUM:=R\_NUM;

SELF.K\_NUM:=K\_NUM;

SELF.F\_NUM:=F\_NUM;

SELF.A\_NUM:=A\_NUM;

SELF.LAIKS:=LAIKS;

RETURN ;

END T\_REKINS;

end;

Objekta tipa TIPS\_NOREKINS izveide:

create type TIPS\_NOREKINS as object(

BANKA varchar2(20),

KONTS varchar2(20)

);

CREATE OR REPLACE TYPE TIPS\_ATSAUCE as object (

ATSAUCE REF T\_REKINS

);

create or replace type KOLEKCIJA\_NOMAS\_VESTURE as table of TIPS\_ATSAUCE;

Tabulas REKINI izveide

create table REKINI of T\_REKINS;

### Tabulas KLIENTI un iesaistīto objektu tipu izveide

Objekta tipa TIPS\_KLIENTS izveide:

create or replace type TIPS\_KLIENTS as object(

K\_NUM number,

K\_UZV varchar2(25),

K\_VAR varchar2(25),

P\_KODS varchar2(15),

VAD\_APLIECIBA date,

BONUSA\_LIMENIS varchar2(15),

NOREKINS TIPS\_NOREKINS,

NOMAS\_VESTURE KOLEKCIJA\_NOMAS\_VESTURE,

constructor function TIPS\_KLIENTS (K\_NUM number, K\_UZV varchar2, K\_VAR varchar2, P\_KODS varchar2, VAD\_APLIECIBA date)return SELF as result,

member function BRAUKSANAS\_PIEREDZE return number

);

Java faila metodes definēšana:

CREATE OR REPLACE JAVA SOURCE NAMED "Pieredze" AS

import java.sql.\*;

import oracle.jdbc.\*;

import java.util.\*;

import java.io.\*;

public class Pieredze{

public static Integer rezultats(Integer vaditajaGgads)

throws SQLException{ return (2020-vaditajaGads);}}

Funkcijas definēšana:

create or replace function ATRAST\_PIEREDZI(vaditajaGads IN NUMBER) return number as

language java name 'Pieredze.rezultats(java.lang.Integer) return Integer';

Type body definēšana:

create or replace type body TIPS\_KLIENTS as

constructor function TIPS\_KLIENTS (K\_NUM number, K\_UZV varchar2, K\_VAR varchar2, P\_KODS varchar2, VAD\_APLIECIBA date)

return SELF as result is

begin

SELF.K\_NUM:=K\_NUM;

SELF.K\_UZV:=K\_UZV;

SELF.K\_VAR:=K\_VAR;

SELF.P\_KODS:=P\_KODS;

SELF.VAD\_APLIECIBA:=VAD\_APLIECIBA;

RETURN;

end TIPS\_KLIENTS;

member function BRAUKSANAS\_PIEREDZE return number IS

skaitlis NUMBER;

BEGIN

skaitlis:= TO\_NUMBER(TO\_CHAR(EXTRACT(YEAR from SELF.VAD\_APLIECIBA)));

RETURN ATRAST\_PIEREDZI(skaitlis);

END BRAUKSANAS\_PIEREDZE;

END;

Tabulas KLIENTI izveide:

create table KLIENTI of TIPS\_KLIENTS

nested table NOMAS\_VESTURE store as TAB\_VESTURE;

### Skata SKATS\_KLIENTI izveide

Skata SKATS\_KLIENTI un tā objekta tipa TIPS\_SKATS\_KLIENTS izveide:

create or replace type TIPS\_SKATS\_KLIENTS as object (

k\_num number,

k\_uzv varchar2(25),

k\_var varchar2(25),

vad\_aplieciba date,

MAP MEMBER function BRAUKSANAS\_STAZS return number

);

create or replace type body TIPS\_SKATS\_KLIENTS as

MAP member function BRAUKSANAS\_STAZS return number IS

BEGIN

RETURN EXTRACT(YEAR FROM SYSDATE)-EXTRACT(YEAR FROM SELF.VAD\_APLIECIBA);

END BRAUKSANAS\_STAZS;

END;

create or replace view SKATS\_KLIENTI of TIPS\_SKATS\_KLIENTS

with object identifier(k\_num) as

select a.k\_num, a.k\_uzv, a.k\_var, a.vad\_aplieciba

FROM klienti A;

Skata SKATS REKINI atteicīgā objekta tipa TIPS\_SKATS\_REKINS izveide:

create or replace type TIPS\_SKATS\_REKINI as object (

R\_NUM number,

K\_NUM number,

F\_NUM number,

A\_NUM number,

CENA number,

LAIKS number,

ATSAUCE\_KLIENTS ref TIPS\_SKATS\_KLIENTS,

member function CIK\_JAMAKSA return NUMBER

);Java funkcijas izveide:

create or replace java source named "SummasAprekins" as

import java.sql.\*;

import oracle.jdbc.\*;

import java.util.\*;

import java.io.\*;

public class SummasAprekins{

public static integer rezultats(integer nomasCena, integer nomasLaiks)

throws SQLException{ return (nomasCena \* nomasLaiks);}}

Funkcijas izveide izmantojot augstāk definēto Java source failu:

create or replace function ATRASTSUMMU(nomasCena in number, nomasLaiks in number) return number as

LANGUAGE JAVA NAME 'SummasAprekins.rezultats(java.lang.integer, java.lang.integer) return integer';

Objekta tipa BODY definēšana:

create or replace type body TIPS\_SKATS\_REKINI as

member function CIK\_JAMAKSA return NUMBER is

BEGIN

RETURN ATRASTSUMMU(SELF.CENA, SELF.LAIKS);

end CIK\_JAMAKSA;

END;

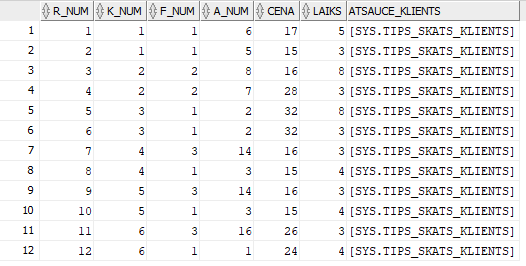
Skata SKATS\_REKINI izveidošana:

create or replace view SKATS\_REKINI of TIPS\_SKATS\_REKINI

with object identifier (R\_NUM) as

SELECT A.R\_NUM, A.K\_NUM, A.F\_NUM, A.A\_NUM, A.CENA, A.LAIKS, MAKE\_REF(SKATS\_KLIENTI, A.K\_NUM) ATSAUCE\_KLIENTS

FROM REKINI A;



2.1.5.1.attēls. Skata SKATS\_REKINI ieraksti

## Datu ievade

### Tabula NOMAS\_FIRMAS

Tabulā NOMAS\_FIRMAS:

insert into NOMAS\_FIRMAS values (TIPS\_AUTONOMAS\_FIRMA(1, 'CARGURU', T\_ADRESE('LATVIJA', 'RIGA', 'KRASTA IELA', 'LV-1000') ));

insert into NOMAS\_FIRMAS values (TIPS\_AUTONOMAS\_FIRMA(2, 'CITY BEE', T\_ADRESE('LATVIJA', 'JURMALA', 'SLOKAS IELA', 'LV-2540') ));

insert into NOMAS\_FIRMAS values (TIPS\_AUTONOMAS\_FIRMA(3, 'FIXSY', T\_ADRESE('LATVIJA', 'RIGA', 'VALMIERAS IELA', 'LV-1040') ));



2.2.1.attēls. Ieraksti tabulā NOMAS\_FIRMAS

update NOMAS\_FIRMAS a

set a.NOMAS\_AUTO =KOLEKCIJA\_AUTO(

T\_AUTO(1,'TOYOTA', 'CHR', 2018, 30000, 24, 10, TO\_DATE('2021/06/14', 'YYYY,MM,DD') ),

T\_AUTO(2,'TOYOTA', 'RAV-4', 2019, 40000, 32, 12, TO\_DATE('2021/09/14', 'YYYY,MM,DD')),

T\_AUTO(3, 'TOYOTA', 'COROLLA', 2015, 100000, 15,30, TO\_DATE('2021/03/15', 'YYYY,MM,DD')),

KRAVAS\_AUTO(4, 'TOYOTA', 'HILUX', 2015, 300000, 18, 30,TO\_DATE('2021/08/14', 'YYYY,MM,DD'), 200),

KRAVAS\_AUTO(5, 'TOYOTA', 'HYT', 2010, 350000, 15, 31, TO\_DATE('2021/06/21', 'YYYY,MM,DD'),230),

KRAVAS\_AUTO(6, 'TOYOTA', 'HYT', 2011,200000, 17, 28, TO\_DATE('2021/07/04', 'YYYY,MM,DD'),180)

)

where a.F\_NUM=1;

update NOMAS\_FIRMAS a

set a.NOMAS\_AUTO =KOLEKCIJA\_AUTO (

KRAVAS\_AUTO(7, 'CHRYSLER', 'DELTA', 2014, 320000, 28, 30,TO\_DATE('2021/10/24', 'YYYY,MM,DD'), 300),

KRAVAS\_AUTO(8, 'SKODA', 'PERT', 2011, 350000, 16, 41, TO\_DATE('2021/07/29', 'YYYY,MM,DD'),240),

KRAVAS\_AUTO(9, 'MERCEDES-BENZ', 'SATA', 2018,200000, 27, 38, TO\_DATE('2021/08/04', 'YYYY,MM,DD'),280),

AUTOBUSS(10, 'ISUZU', 'KATA', 2014, 32000, 38, 18,TO\_DATE('2021/11/24', 'YYYY,MM,DD'), 30),

AUTOBUSS(11, 'MERCEDES-BENZ', 'SANA', 2020, 35000, 36, 11, TO\_DATE('2021/10/29', 'YYYY,MM,DD'),24),

AUTOBUSS(12, 'MERCEDES-BENZ', 'SANA', 2018,200000, 17, 38, TO\_DATE('2021/08/04', 'YYYY,MM,DD'),24),

T\_AUTO(19,'FIAT', 'ATTA', 2019, 30000, 34, 15, TO\_DATE('2021/06/14', 'YYYY,MM,DD') ),

T\_AUTO(20,'FIAT', 'VERANO', 2018, 40000, 32, 23, TO\_DATE('2021/05/14', 'YYYY,MM,DD')),

T\_AUTO(21, 'FIAT', 'SOMBRERO', 2015, 100000, 19,32, TO\_DATE('2021/03/24', 'YYYY,MM,DD'))

)

where a.F\_NUM=2;

update NOMAS\_FIRMAS a

set a.NOMAS\_AUTO =KOLEKCIJA\_AUTO(

AUTOBUSS(13, 'ISUZU', 'KATA', 2014, 32000, 35, 18,TO\_DATE('2021/07/20', 'YYYY,MM,DD'), 46),

AUTOBUSS(14, 'ISUZU', 'IKIANA', 2010, 350670, 16, 41, TO\_DATE('2021/03/19', 'YYYY,MM,DD'),34),

AUTOBUSS(15, 'ISUZU', 'KATA', 2014,200000, 27, 38, TO\_DATE('2021/08/04', 'YYYY,MM,DD'),38),

KRAVAS\_AUTO(16, 'CHRYSLER', 'DELTA', 2012, 326000, 26, 30,TO\_DATE('2021/09/24', 'YYYY,MM,DD'), 260),

KRAVAS\_AUTO(17, 'SKODA', 'PERT', 2011, 359800, 18, 31, TO\_DATE('2021/07/13', 'YYYY,MM,DD'),220),

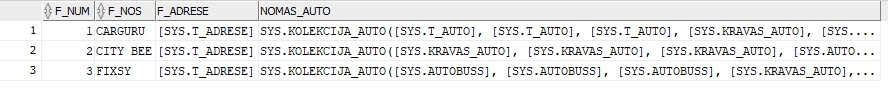
KRAVAS\_AUTO(18, 'MERCEDES-BENZ', 'SATA', 2016,287000, 25, 38, TO\_DATE('2021/08/04', 'YYYY,MM,DD'),280),

T\_AUTO(22,'BMW', 'DUST', 2019, 33400, 34, 28, TO\_DATE('2021/06/18', 'YYYY,MM,DD') ),

T\_AUTO(23,'BMW', 'M7', 2018, 40000, 32, 31, TO\_DATE('2021/05/21', 'YYYY,MM,DD')),

T\_AUTO(24, 'BMW', 'A7', 2016, 100000, 19,42, TO\_DATE('2021/04/23', 'YYYY,MM,DD'))

)

where a.F\_NUM=3;

### Tabula KLIENTI

Rēķinu atsauču ievietošana tabulā KLIENTI:

DECLARE

ats1 ref t\_rekins;

ats2 ref t\_rekins;

begin

select ref(a) into ats1 from REKINI a where a.k\_num=2 and a.r\_num=3;

select ref(a) into ats2 from REKINI a where a.k\_num=2 and a.r\_num=4;

update KLIENTI B

set B.NOMAS\_VESTURE=KOLEKCIJA\_NOMAS\_VESTURE(TIPS\_ATSAUCE(ATS1), TIPS\_ATSAUCE(ATS2))

WHERE B.K\_NUM=2;

end;

Ierakstu veikšana tabulā KLIENTI:

create sequence NUMURI\_2

start with 1

increment by 1

order;

DECLARE

i number;

type masiva\_tips\_uzvardi is varray(500) of varchar2(30);

masivs\_uzvards masiva\_tips\_uzvardi:=masiva\_tips\_uzvardi('SMITH', 'JOHNSON', 'WILLIAMS', 'JONES', 'BROWN', 'DAVIS', 'MILLER', 'WILSON', 'MOORE', 'TAYLOR', 'ANDERSON', 'THOMAS', 'JACKSON', 'WHITE', 'HARRIS', 'MARTIN', 'THOMPSON', 'GARCIA', 'MARTINEZ', 'ROBINSON', 'CLARK', 'RODRIGUEZ', 'LEWIS', 'LEE', 'WALKER', 'HALL', 'ALLEN', 'YOUNG', 'HERNANDEZ', 'KING', 'WRIGHT', 'LOPEZ', 'HILL', 'SCOTT', 'GREEN', 'ADAMS', 'BAKER', 'GONZALEZ', 'NELSON', 'CARTER', 'MITCHELL', 'PEREZ', 'ROBERTS', 'TURNER', 'PHILLIPS', 'CAMPBELL', 'PARKER', 'EVANS', 'EDWARDS', 'COLLINS', 'STEWART','SANCHEZ', 'MORRIS', 'ROGERS', 'REED', 'COOK', 'MORGAN', 'BELL', 'MURPHY', 'BAILEY', 'RIVERA', 'COOPER', 'RICHARDSON','COX', 'HOWARD', 'WARD', 'TORRES', 'PETERSON', 'GRAY', 'RAMIREZ', 'JAMES', 'WATSON', 'BROOKS', 'KELLY', 'SANDERS', 'PRICE', 'BENNETT', 'WOOD', 'BARNES', 'ROSS', 'HENDERSON', 'COLEMAN', 'JENKINS', 'PERRY', 'POWELL', 'LONG', 'PATTERSON', 'HUGHES', 'FLORES', 'WASHINGTON', 'BUTLER', 'SIMMONS', 'FOSTER', 'GONZALES', 'BRYANT', 'ALEXANDER', 'RUSSELL', 'GRIFFIN', 'DIAZ', 'HAYES', 'MYERS', 'FORD', 'HAMILTON', 'GRAHAM', 'SULLIVAN', 'WALLACE', 'WOODS', 'COLE', 'WEST', 'JORDAN', 'OWENS', 'REYNOLDS', 'FISHER', 'ELLIS', 'HARRISON', 'GIBSON', 'MCDONALD', 'CRUZ','MARSHALL', 'ORTIZ', 'GOMEZ', 'MURRAY', 'FREEMAN', 'WELLS', 'WEBB', 'SIMPSON', 'STEVENS', 'TUCKER', 'PORTER','HUNTER', 'HICKS', 'CRAWFORD', 'HENRY', 'BOYD', 'MASON', 'MORALES', 'KENNEDY', 'WARREN', 'DIXON', 'RAMOS', 'REYES', 'BURNS', 'GORDON', 'SHAW', 'HOLMES', 'RICE', 'ROBERTSON', 'HUNT', 'BLACK', 'DANIELS', 'PALMER', 'MILLS', 'NICHOLS', 'GRANT', 'KNIGHT', 'FERGUSON', 'ROSE', 'STONE', 'HAWKINS', 'DUNN', 'PERKINS', 'HUDSON', 'SPENCER', 'GARDNER', 'STEPHENS', 'PAYNE', 'PIERCE', 'BERRY', 'MATTHEWS', 'ARNOLD', 'WAGNER', 'WILLIS', 'RAY', 'WATKINS','OLSON', 'CARROLL', 'DUNCAN', 'SNYDER', 'HART', 'CUNNINGHAM', 'BRADLEY', 'LANE', 'ANDREWS', 'RUIZ', 'HARPER', 'FOX', 'RILEY', 'ARMSTRONG', 'CARPENTER', 'WEAVER', 'GREENE', 'LAWRENCE', 'ELLIOTT', 'CHAVEZ', 'SIMS', 'AUSTIN', 'PETERS', 'KELLEY', 'FRANKLIN', 'LAWSON', 'FIELDS', 'GUTIERREZ', 'RYAN', 'SCHMIDT', 'CARR', 'VASQUEZ', 'CASTILLO','WHEELER', 'CHAPMAN', 'OLIVER', 'MONTGOMERY', 'RICHARDS', 'WILLIAMSON', 'JOHNSTON', 'BANKS', 'MEYER', 'BISHOP', 'MCCOY', 'HOWELL', 'ALVAREZ', 'MORRISON', 'HANSEN', 'FERNANDEZ', 'GARZA', 'HARVEY', 'LITTLE', 'BURTON', 'STANLEY', 'NGUYEN', 'GEORGE', 'JACOBS', 'REID', 'KIM', 'FULLER', 'LYNCH', 'DEAN', 'GILBERT', 'GARRETT', 'ROMERO', 'WELCH', 'LARSON', 'FRAZIER', 'BURKE', 'HANSON', 'DAY', 'MENDOZA', 'MORENO', 'BOWMAN', 'MEDINA', 'FOWLER', 'BREWER', 'HOFFMAN', 'CARLSON', 'SILVA', 'PEARSON', 'HOLLAND', 'DOUGLAS', 'FLEMING', 'JENSEN', 'VARGAS', 'BYRD', 'DAVIDSON', 'HOPKINS', 'MAY', 'TERRY', 'HERRERA', 'WADE', 'SOTO', 'WALTERS', 'CURTIS', 'NEAL', 'CALDWELL', 'LOWE', 'JENNINGS', 'BARNETT', 'GRAVES', 'JIMENEZ', 'HORTON', 'SHELTON', 'BARRETT', 'OBRIEN', 'CASTRO', 'SUTTON', 'GREGORY', 'MCKINNEY','LUCAS', 'MILES', 'CRAIG', 'RODRIQUEZ', 'CHAMBERS', 'HOLT', 'LAMBERT', 'FLETCHER', 'WATTS', 'BATES', 'HALE', 'RHODES', 'PENA', 'BECK', 'NEWMAN', 'HAYNES', 'MCDANIEL', 'MENDEZ', 'BUSH', 'VAUGHN', 'PARKS', 'DAWSON', 'SANTIAGO','NORRIS', 'HARDY', 'LOVE', 'STEELE', 'CURRY', 'POWERS', 'SCHULTZ', 'BARKER', 'GUZMAN', 'PAGE', 'MUNOZ', 'BALL','KELLER', 'CHANDLER', 'WEBER', 'LEONARD', 'WALSH', 'LYONS', 'RAMSEY', 'WOLFE', 'SCHNEIDER', 'MULLINS', 'BENSON', 'SHARP', 'BOWEN', 'DANIEL', 'BARBER', 'CUMMINGS', 'HINES', 'BALDWIN', 'GRIFFITH', 'VALDEZ', 'HUBBARD', 'SALAZAR', 'REEVES', 'WARNER', 'STEVENSON', 'BURGESS', 'SANTOS', 'TATE', 'CROSS', 'GARNER', 'MANN', 'MACK', 'MOSS', 'THORNTON', 'DENNIS', 'MCGEE', 'FARMER', 'DELGADO', 'AGUILAR', 'VEGA', 'GLOVER', 'MANNING', 'COHEN', 'HARMON', 'RODGERS', 'ROBBINS', 'NEWTON', 'TODD', 'BLAIR', 'HIGGINS', 'INGRAM', 'REESE', 'CANNON', 'STRICKLAND', 'TOWNSEND', 'POTTER', 'GOODWIN', 'WALTON', 'ROWE', 'HAMPTON', 'ORTEGA', 'PATTON', 'SWANSON', 'JOSEPH', 'FRANCIS', 'GOODMAN', 'MALDONADO', 'YATES', 'BECKER', 'ERICKSON', 'HODGES', 'RIOS', 'CONNER', 'ADKINS', 'WEBSTER', 'NORMAN', 'MALONE', 'HAMMOND', 'FLOWERS', 'COBB', 'MOODY', 'QUINN', 'BLAKE', 'MAXWELL', 'POPE', 'FLOYD', 'OSBORNE', 'PAUL', 'MCCARTHY', 'GUERRERO', 'LINDSEY', 'ESTRADA', 'SANDOVAL', 'GIBBS', 'TYLER', 'GROSS', 'FITZGERALD', 'STOKES', 'DOYLE', 'SHERMAN', 'SAUNDERS', 'WISE', 'COLON', 'GILL', 'ALVARADO', 'GREER', 'PADILLA', 'SIMON', 'WATERS', 'NUNEZ', 'BALLARD', 'SCHWARTZ', 'MCBRIDE', 'HOUSTON', 'CHRISTENSEN', 'KLEIN', 'PRATT', 'BRIGGS', 'PARSONS', 'MCLAUGHLIN', 'ZIMMERMAN', 'FRENCH', 'BUCHANAN', 'MORAN', 'COPELAND', 'ROY', 'PITTMAN', 'BRADY', 'MCCORMICK', 'HOLLOWAY', 'BROCK', 'POOLE', 'FRANK', 'LOGAN', 'OWEN', 'BASS', 'MARSH', 'DRAKE', 'WONG', 'JEFFERSON', 'PARK', 'MORTON', 'ABBOTT', 'SPARKS', 'PATRICK', 'NORTON', 'HUFF', 'CLAYTON', 'MASSEY', 'LLOYD', 'FIGUEROA', 'CARSON', 'BOWERS', 'ROBERSON', 'BARTON', 'TRAN', 'LAMB', 'HARRINGTON', 'CASEY', 'BOONE', 'CORTEZ', 'CLARKE', 'MATHIS', 'SINGLETON', 'WILKINS', 'CAIN', 'BRYAN', 'UNDERWOOD', 'HOGAN', 'MCKENZIE','COLLIER', 'LUNA', 'PHELPS', 'MCGUIRE', 'ALLISON', 'BRIDGES', 'WILKERSON', 'NASH', 'SUMMERS', 'ATKINS');

type masiva\_tips\_vardi is varray(120) of varchar2(20);

masivs\_vardi masiva\_tips\_vardi:=masiva\_tips\_vardi(

'Janis','Juris', 'Julijs','Indulis','Artūrs', 'Agris', 'Peteris', 'Arnolds', 'Ivars', 'Varis','Gunars','Valdis', 'Ingmārs', 'Aloīzs', 'Adams', 'Ventis', 'Kārlis', 'Stefans', 'Ābrams', 'Joske', 'Gatis', 'Pavils', 'Fīlips', 'Donalds', 'Bilis', 'Mārtins', 'Kristaps', 'Viesturs', 'Atis', 'Ainārs','Bonus', 'Didzis', 'Druvis', 'Ēriks', 'Eidis', 'Elmars', 'Ernests', 'Ervīns', 'Einārs', 'Ilmārs','Jekabs', 'Gunis', 'Harijs', 'Ludvigs', 'Modris', 'Māris', 'Mikus', 'Renārs', 'Odisejs', 'Sīmanis', 'Monvids', 'Maksis', 'Alnis', 'Cinis', 'Elvis', 'Erkils', 'Mikus', 'Mintauts', 'Rolands', 'Raivis','Maija','Mirdza', 'Zane', 'Inese', 'Rasma', 'Anita', 'Gundega', 'Dziesma', 'Alīda', 'Gunta', 'Sonita', 'Sandra', 'Liene', 'Marija', 'Izabella','Andra', 'Madara', 'Mirdza', 'Maira', 'Gita', 'Vita', 'Mara', 'Jūlija', 'Esmeralda', 'Karīna', 'Arnita', 'Anna', 'Ieva', 'Linda', 'Vera','Minna', 'Maruta', 'Maiga', 'Melānija', 'Naura', 'Nora', 'Natālija', 'Nanīna', 'Nelma', 'Ņina', 'Maira', 'Malvīna', 'Marga', 'Cidonija', 'Begonija', 'Acālija', 'Eiženija', 'Biruta', 'Aldona', 'Agate', 'Alma','Asnete', 'Antrese', 'Betija', 'Kasandra', 'Karlīna', 'Lauma', 'Laima', 'Linda', 'Lora');

BEGIN

insert into KLIENTI values (

TIPS\_KLIENTS(1, MASIVS\_UZVARDS(TRUNC(DBMS\_RANDOM.VALUE(1,500))), MASIVS\_VARDI(TRUNC(DBMS\_RANDOM.VALUE(1,120))),'090392-11923', TO\_DATE('2011/07/20', 'YYYY,MM,DD')));

insert into KLIENTI values (

TIPS\_KLIENTS(2, MASIVS\_UZVARDS(TRUNC(DBMS\_RANDOM.VALUE(1,500))), MASIVS\_VARDI(TRUNC(DBMS\_RANDOM.VALUE(1,120))),'121279-11945', TO\_DATE('2001/07/20', 'YYYY,MM,DD')));

insert into KLIENTI values (

TIPS\_KLIENTS(3, MASIVS\_UZVARDS(TRUNC(DBMS\_RANDOM.VALUE(1,500))), MASIVS\_VARDI(TRUNC(DBMS\_RANDOM.VALUE(1,120))),'190390-11653', TO\_DATE('2008/04/20', 'YYYY,MM,DD')));

insert into KLIENTI values (

TIPS\_KLIENTS(4, MASIVS\_UZVARDS(TRUNC(DBMS\_RANDOM.VALUE(1,500))), MASIVS\_VARDI(TRUNC(DBMS\_RANDOM.VALUE(1,120))),'130790-11456', TO\_DATE('2009/07/21', 'YYYY,MM,DD')));

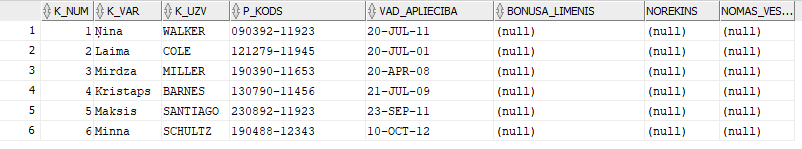
insert into KLIENTI values (

TIPS\_KLIENTS(5, MASIVS\_UZVARDS(TRUNC(DBMS\_RANDOM.VALUE(1,500))), MASIVS\_VARDI(TRUNC(DBMS\_RANDOM.VALUE(1,120))),'230892-11923', TO\_DATE('2011/09/23', 'YYYY,MM,DD')));

insert into KLIENTI values (

TIPS\_KLIENTS(6, MASIVS\_UZVARDS(TRUNC(DBMS\_RANDOM.VALUE(1,500))), MASIVS\_VARDI(TRUNC(DBMS\_RANDOM.VALUE(1,120))),'190488-12343', TO\_DATE('2012/10/10', 'YYYY,MM,DD')));

END;



update KLIENTI A

set a.NOREKINS=TIPS\_NOREKINS('SWEDBANK', 'LV81HABA1234');

insert into REKINI VALUES(T\_REKINS(1,1,1,6,5));

insert into REKINI VALUES(T\_REKINS(2,1,1,5,3));

insert into REKINI VALUES(T\_REKINS(3,2,2,8,8));

insert into REKINI VALUES(T\_REKINS(4,2,2,7,3));

insert into REKINI VALUES(T\_REKINS(5,3,1,2,8));

insert into REKINI VALUES(T\_REKINS(6,3,1,2,3));

insert into REKINI VALUES(T\_REKINS(7,4,3,14,3));

insert into REKINI VALUES(T\_REKINS(8,4,1,3,4));

insert into REKINI VALUES(T\_REKINS(9,5,3,14,3));

insert into REKINI VALUES(T\_REKINS(10,5,1,3,4));

insert into REKINI VALUES(T\_REKINS(11,6,3,16,3));

insert into REKINI VALUES(T\_REKINS(12,6,1,1,4));

Cenas ielikšana rēķinā izmantojot SKATS\_NOMAS\_AUTO

DECLARE

ATS1 NUMBER;

i NUMBER;

K NUMBER;

BEGIN

for i in 1..12 loop

SELECT B.A\_NUM INTO K FROM REKINI B WHERE B.R\_NUM=i;

SELECT A.CENA INTO ATS1 FROM skats\_nomas\_auto A where a.a\_num=k;

update REKINI c

set c.cena=ats1

where c.R\_NUM=i;

end loop;

END;

# VAICĀJUMI UN METOŽU IZMANTOŠANA

## Metode IZVADE

Metode definēta tipam TIPS\_NOMASFIRMA.

Vaicājums:

set serveroutput on size 5000;

declare piemers\_1 TIPS\_AUTONOMAS\_FIRMA;

BEGIN

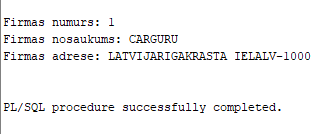
SELECT VALUE(A) INTO PIEMERS\_1

FROM NOMAS\_FIRMAS A

WHERE VALUE(A).F\_NUM=1;

PIEMERS\_1.IZVADE;

END;

Vaicājuma rezultāts:

## Metode PARBAUDE

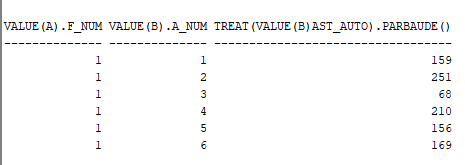
Metode PARBAUDE ir definēta tipam T\_AUTO un tā apakštipi to pārraksta.

Vaicājums, kur pārbaudes laiku aprēķina kā vieglajai automašīnai:

select value(a).F\_NUM, VALUE(B).A\_NUM, TREAT (VALUE(B) AS T\_AUTO).PARBAUDE()

from NOMAS\_FIRMAS A, TABLE(VALUE(A).NOMAS\_AUTO) B

WHERE F\_NUM=1;

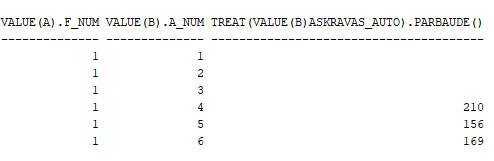
Vaicājuma rezultāti:

Vaicājums, kur pārbaudes datumu aprēķina kravas auto:

select value(a).F\_NUM, VALUE(B).A\_NUM, TREAT (VALUE(B) AS KRAVAS\_AUTO).PARBAUDE()

from NOMAS\_FIRMAS A, TABLE(VALUE(A).NOMAS\_AUTO) B

WHERE F\_NUM=1;

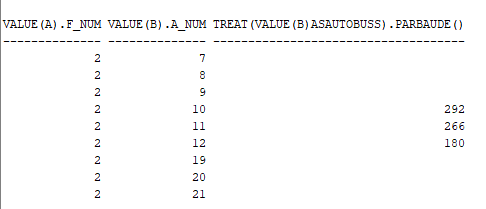
Vaicājuma rezultāti:

Vaicājums, kur pārbaudes datumu aprēķina autobusiem:

select value(a).F\_NUM, VALUE(B).A\_NUM, TREAT (VALUE(B) AS AUTOBUSS).PARBAUDE()

from NOMAS\_FIRMAS A, TABLE(VALUE(A).NOMAS\_AUTO) B

WHERE F\_NUM=2;

Vaicājuma rezultāti:

## Metode SALIDZINAT\_NOMAS\_CENU

Metode definēta tipam TIPS\_NOMAS\_AUTO

Vaicājums, kas salīdzina nomas cenas kravas automašinām:

select value(a).A\_NUM, VALUE(A).MARKA, vALUE(A).MODELIS, VALUE(A).K\_IETILPIBA,

VALUE(A).CENA, VALUE(A).SALIDZINAT\_NOMAS\_CENU(C.OBJ) AS SALIDZINAJUMS

FROM SKATS\_NOMAS\_AUTO A,

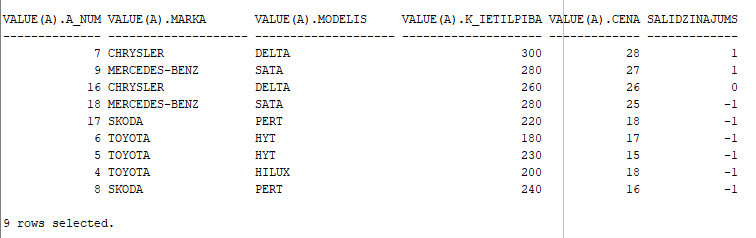
(SELECT VALUE(B) AS OBJ

FROM SKATS\_NOMAS\_AUTO B

WHERE VALUE(B).A\_NUM=16)C

WHERE VALUE(A).K\_IETILPIBA>0

ORDER BY SALIDZINAJUMS DESC;

Vaicājuma rezultāti:

## MAP Metode BRAUKSANAS\_STAZS

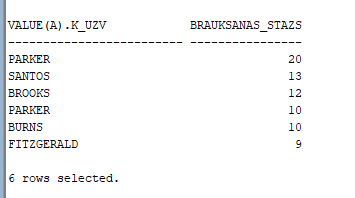
Metode ir definēta tipam TIPS\_SKATS\_KLIENTS.

Vaicājums, kas sakārto klientus pēc to braukšanas stāža:

select value(a).K\_UZV, VALUE(A).BRAUKSANAS\_STAZS() AS BRAUKSANAS\_STAZS

FROM SKATS\_KLIENTI A

ORDER BY BRAUKSANAS\_STAZS DESC;

Vaicājuma Rezultāti:

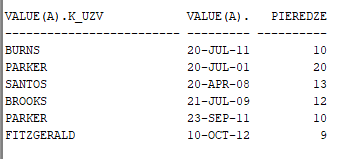
## Member metode BRAUKSANAS\_PIEREDZE java

Metode ir definēta tipam TIPS\_KLIENTS un ir realizēta ar JAVA palīdzību.

Vaicājums:

select value(a).K\_UZV, value(a).VAD\_APLIECIBA, value(a).brauksanas\_pieredze()as PIEREDZE

from KLIENTI a;

Vaicājuma rezultāti:

## Metode CIK\_JAMAKSA

Metode ir definēta tipam TIPS\_SKATS\_KLIENTI un tiek izmantots objektu skats SKATS\_REKINI.

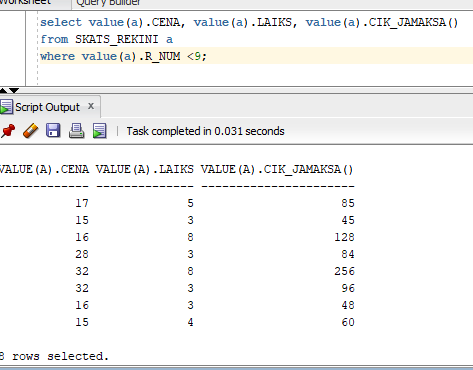
Vaicājums:

select value(a).cena, value(a).laiks, value(a).cik\_jamaksa()

from skats\_rekini a

where value(a).r\_num <9;

Vaicājuma rezultāti:



# Secinājumi

Ceturtā praktiskā darba ietvaros ietilpa izveidot divas vai trīs relāciju-objektu tabulas, ar salīdzinoši sarežģītu shēmu. Papildus ierastajai objektu tipu veidošanai, bija nepieciešams izveidot objekta tipu hierarhiju. Šajā gadījumā tas tika realizēts ar dažādu automašīnu izveidošanu. Katrai automašīnai bija sava PĀRBAUDES aprēķina metode, kas atgrieza dienu skaitu, kad nepieciešams veikt pārbaudi pirms skates beigām. Te arī tika realizēta šīs metodes pārrakstīšana pie katra apakš tipa . Līdzīgi, kā trešā darba ietvaros, bija jāizveido objektu skati. Šajā darbā objekta skati tika izveidotu lai sasaistītu informāciju par automašīnu nomu un rēķiniem. Tika izveidots arī salikts skats, kas atgriež informāciju par klientiem un to rēķiniem. Objekta skatiem tika izveidotas arī metodes. Šīs metodes šoreiz nebija ar augstāku komplicētību. Darbā tika izveidotas arī divas vienkāršas metodes ar JAVA palīdzību. Neraugoties uz realizēto metožu vienkāršību, var saprast, ka šādā veidā ir iespējams realizēt dažādu sarežģītību metodes, kā arī relāciju-objektu tabulu lietderību, jo salīdzinot ar tradicionālajām relāciju tabulām, vaicājumu ātrumi objektu tabulās ir ar lēnāku izpildi. Dažādu metožu izveidošanas iespējas skaidri norāda uz relāciju-objektu datu bāzes priekšrocībām. Kopsummā ceturtā darba nospraustie uzdevumi ir realizēti apmierinoši ( ne visas darbības atskaites dokumentā ir atspoguļotas to izpildes secībā).

Kopsummā šī mācību kursā bija iespēja samērā augstā līmenī apgūt relāciju-objektu datu bāzes pamatus.