

RĪGAS TEHNISKĀ UNIVERSITĀTE

Datorzinātnes un informācijas tehnoloģijas fakultāte

mācību priekšmetā

“Lielu datu bāzu tehnoloģijas (DB2)”

3.praktiskais darbs

‘Objektu metožu izmantošana relāciju-objektu datu bāzē’

Izstrādāja: Jolanta Bērziņa

Stud.apl.nr 181RDB178

2020./2021. m.g.

**Saturs**

[Ievads 2](#_Toc58331938)

[1. Objektu tabulu izveide 3](#_Toc58331939)

[1.1. Tabulu ierakstu shēmas attainojums 3](#_Toc58331940)

[1.2. Objektu tipu shēmu un metožu definēšana 4](#_Toc58331941)

[2. Tabulu aizpildīšana un metožu izmantošana 11](#_Toc58331942)

[2.1. Constructor function pielietojums aizpildot tabulas ar ierakstiem 11](#_Toc58331943)

[2.1.1. Tabula Automasinas 11](#_Toc58331944)

[2.1.2. Tabula Autonomas\_firmas 12](#_Toc58331945)

[2.1.3. Tabula KLIENTI 13](#_Toc58331946)

[2.1.4. Tabula Rekini 16](#_Toc58331947)

[2.2. Member procedure un function pielietojums 17](#_Toc58331948)

[2.3. ORDER funkcijas pielietojums 19](#_Toc58331949)

[2.4. MAP funkcijas pielietojums 21](#_Toc58331950)

[3. Objekta skata izveide, kas ietver MEMBER metodi 22](#_Toc58331951)

[Secinājumi 27](#_Toc58331952)

# Ievads

Trešais prasktiskais darbs mācību priekšmetā ‘Lielo datu bāzu tehnoloģijas’ iekļauj mērķi apģut objektu metožu realizāciju un pievienošanu objektu struktūrām. Praktiskajā darbā jāveido saliktas objektu tabulas (rindas objekti iekļauj pakārtotos objektus un pakārtotas objektu kolekcijas), objektu tipos iekļaujot objektu metožu definējumus. Līdz ar to šajā darbā ir jādefinē sekojošas metodes:

1. Objekta konstruktora metode;
2. Objektu sakārtošanas metode (MAP Member);
3. Objektu salīdzināšanas metode (ORDER Member);
4. Universāla tipa metode (MEMBER).

Tāpat kā iepriekšējos praktiskajos darbos, tabulās ir jāievada testa dati (**nav** obligāti jāģenerē 1000 ieraksti) un jārealizē datu sakārtošana un izgūšana, izmantojot definētās objektu metodes.

Jāizveido objektu skats, kuram arī jādefinē universāla tipa metode (MEMBER). Jāveic datu izgūšana no objekta skata izmantojot metodi.

Praktiskā darba beigās ir jāveic metožu izmantošanas iespēju uzvērtējums.

# Objektu tabulu izveide

## Tabulu ierakstu shēmas attainojums

REF\_AUTO

REF\_AUTO

NOMAS\_MASINAS

NR-IELA

IELA

PILSETA

F\_ADRESE

F\_NUM

F\_NOS

DIB\_GADS

AUTONOMAS\_FIRMA

TIPS\_AUTO

A\_NUM

F\_NUM

MARKA

GADS

MODELIS

A\_MARKA

NOBRAUKUMS

NOMAS\_CENA

NOMAS\_SKAITS

REF\_REKINS

REF\_REKINS

NOMAS\_VESTURE

NOREKINS

KONTA\_NR

BANKA

K\_NUM

K\_VAR

K\_VECUMS

TIPS\_KLIENTS

K\_UZV

TIPS\_REKINS

R\_NUM

K\_NUM

F\_NUM

DATUMS

CENA

LAIKS

GALA\_SUMMA

REF\_AUTO

Šajā darbā tiks attainotas datu struktūras par autonomas firmām, tām piederošajām automašīnām, klientiem, un to rēķinu vēsturi.

## Objektu tipu shēmu un metožu definēšana

Tipa TIPS\_AUTO izveide un tā iekļauto objektu tipu izveide, iekļaujot CONSTRUCTOR function un ORDER member.

TIPS MARKA izvede, kuram ir definēta arī CONSTRUCTOR funkcija:

crate or replace type MARKA as object (

A\_MARKA varchar2(20),

MODELIS varchar2(25),

GADS number,

constructor function MARKA (A\_MARKA varchar2, MODELIS varchar2, GADS number) return SELF as result

);

create or replace type body MARKA as

constructor function MARKA (A\_MARKA varchar2, MODELIS varchar2, GADS number)

return SELF as result is

begin

SELF.A\_MARKA:=A\_MARKA;

SELF.MODELIS:=MODELIS;

SELF.GADS:=GADS;

return;

end;

end;

TIPS\_AUTO izveide, kuram ir definētas divas CONSTRUCTOR funkcijas ar atšķirīgu atribūtu skaitu un ORDER member funkcija, kas salīdzina nomā pieejamo automašinu cenas likmi stundā.

create or replace type TIPS\_AUTO as object(

A\_NUM number,

F\_NUM number,

A\_MARKA MARKA,

NOMAS\_SKAITS number,

NOBRAUKUMS number,

NOMAS\_CENA number,

constructor function TIPS\_AUTO (A\_NUM number,F\_NUM number, A\_MARKA MARKA, NOMAS\_SKAITS number, NOBRAUKUMS number,

NOMAS\_CENA number) return SELF as result,

constructor function TIPS\_AUTO (A\_NUM number,F\_NUM number, A\_MARKA MARKA, NOBRAUKUMS number) return SELF as result,

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

);

create or replace type body TIPS\_AUTO as

constructor function TIPS\_AUTO (A\_NUM number, F\_NUM number, A\_MARKA MARKA, NOMAS\_SKAITS number, NOBRAUKUMS number,

NOMAS\_CENA number) return SELF as result IS

begin

SELF.A\_NUM:=A\_NUM;

SELF.F\_NUM:=F\_NUM;

SELF.A\_MARKA:=A\_MARKA;

SELF.NOMAS\_SKAITS:=NOMAS\_SKAITS;

SELF.NOBRAUKUMS:=NOBRAUKUMS;

SELF.NOMAS\_CENA:=NOMAS\_CENA;

return;

end;

constructor function TIPS\_AUTO (A\_NUM number, F\_NUM number, A\_MARKA MARKA,

NOBRAUKUMS number) return SELF as result is

begin

SELF.A\_NUM:=A\_NUM;

SELF.F\_NUM:=F\_NUM;

SELF.A\_MARKA:=A\_MARKA;

SELF.NOBRAUKUMS:=NOBRAUKUMS;

return;

end;

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

begin

if SELF.NOMAS\_CENA < m\_obj.NOMAS\_CENA then return -1;

elsif SELF.NOMAS\_CENA > m\_obj.NOMAS\_CENA then return 1;

else return 0;

end if;

end SALIDZINAT\_NOMAS\_CENU;

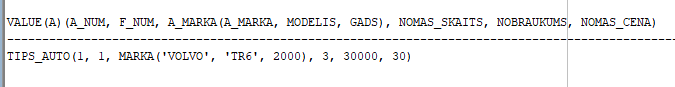
end;

Tabulas AUTOMASINAS izveide no TIPS\_AUTO un pirmo testdatu ievade un izvade ar vaicājumu:

create table AUTOMASINAS of TIPS\_AUTO;

INSERT INTO AUTOMASINAS VALUES (TIPS\_AUTO (1, 1, MARKA('VOLVO', 'TR6', 2000), 3, 30000,30));

select VALUE(a)

from AUTOMASINAS a;

1.2.1.attēls. Tabulas AUTOMASINAS ievaddatu izvade ar vaicājumu

Objekta tipa TIPS\_REKINS izveide, kurā tiek iekļautas CONSTRUCTOR un MEMBER funkcijas. MEMBER funkcija atgriež gala summu, par iznomāto auto, balstoties uz nomas auto cenu stundā un izmantotolaiku.

create or replace type TIPS\_REKINS as object (

R\_NUM number,

K\_NUM number,

F\_NUM number,

DATUMS date,

CENA number,

LAIKS real,

GALA\_SUMMA real,

REF\_AUTO REF TIPS\_AUTO,

constructor function TIPS\_REKINS (R\_NUM number, K\_NUM number, F\_NUM number, DATUMS date, CENA number, LAIKS real)

return SELF as result,

MEMBER function CIK\_JAMAKSA return real

);

create or replace type body TIPS\_REKINS as

constructor function TIPS\_REKINS (R\_NUM number, K\_NUM number, F\_NUM number, DATUMS date, CENA number, LAIKS real)

return SELF as result is

begin

SELF.R\_NUM:=R\_NUM;

SELF.K\_NUM:=K\_NUM;

SELF.F\_NUM:=F\_NUM;

SELF.DATUMS:=DATUMS;

SELF.CENA:=CENA;

SELF.LAIKS:=LAIKS;

Return;

end;

MEMBER function CIK\_JAMAKSA return real is

begin

return (SELF.CENA \* SELF.LAIKS);

end CIK\_JAMAKSA;

end;

Tabulas REKINI izveide:

create table REKINI of TIPS\_REKINS;

Objekta tipa TIPS\_KLIENTS izveide un tajā ietilpstošo tipu objektu izveide, kas ietver CONSTRUCTOR function un MAP member funkcijas.

TIPS\_NOREKINS izveide, kas ietversevī informāciju par klienta banku un kontu, kuram ir definēta arī constructor funkcija.

create or replace type TIPS\_NOREKINS as object (

BANKA varchar2(30),

KONTA\_NR varchar2(12),

constructor function TIPS\_NOREKINS (BANKA varchar2, KONTA\_NR varchar2) return SELF as result

);

create or replace type body TIPS\_NOREKINS as

constructor function TIPS\_NOREKINS (BANKA varchar2, KONTA\_NR varchar2) return SELF as result is

begin

SELF.BANKA:=BANKA;

SELF.KONTA\_NR:=KONTA\_NR;

return;

end;

end;

REF\_REKINS objekta tipa izveide, kas satur atsaci uz rēķinu.

create or replace type REF\_REKINS as object (

ATSAUCE REF TIPS\_REKINS

);

Objekta tips NOMAS\_VESTURE, kas satur visas klienta rēķinu atsauces par nomātām automašīnām.

create or replace type NOMAS\_VESTURE as table of REF\_REKINS;

Paša TIPS\_KLIENTS izveide ar divām CONSTRUCTOR funkcijām, Member veida funkciju, kas izvada informāciju par klientu un MAP member funkciju, kas aprēķina klientu braukšanas pieredzi jeb stāžu.

create or replace type TIPS\_KLIENTS as object (

K\_NUM number,

K\_VAR varchar2(25),

K\_UZV varchar2(30),

K\_VECUMS number,

NOREKINS TIPS\_NOREKINS,

VESTURE\_NOMA NOMAS\_VESTURE,

constructor function TIPS\_KLIENTS (K\_NUM number, K\_VAR varchar2, K\_UZV varchar2, K\_VECUMS number, NOREKINS TIPS\_NOREKINS)

return self as result,

constructor function TIPS\_KLIENTS (K\_NUM number, K\_VAR varchar2, K\_UZV varchar2, K\_VECUMS number) return self as result,

MAP member function BRAUKSANAS\_PIEREDZE return NUMBER,

MEMBER procedure IZVADE

);

create or replace type body TIPS\_KLIENTS as

constructor function TIPS\_KLIENTS (K\_NUM number, K\_VAR varchar2, K\_UZV varchar2, K\_VECUMS number, NOREKINS TIPS\_NOREKINS)

return self as result is

begin

SELF.K\_NUM:=K\_NUM;

SELF.K\_VAR:=K\_VAR;

SELF.K\_UZV:=K\_UZV;

SELF.K\_VECUMS:=K\_VECUMS;

SELF.NOREKINS:=NOREKINS;

RETURN;

end;

constructor function TIPS\_KLIENTS (K\_NUM number, K\_VAR varchar2, K\_UZV varchar2, K\_VECUMS number)

return self as result is

begin

SELF.K\_NUM:=K\_NUM;

SELF.K\_VAR:=K\_VAR;

SELF.K\_UZV:=K\_UZV;

SELF.K\_VECUMS:=K\_VECUMS;

RETURN;

end;

MAP member function BRAUKSANAS\_PIEREDZE return number is

begin

return SELF.K\_VECUMS-18;

end BRAUKSANAS\_PIEREDZE;

MEMBER procedure IZVADE is

begin

dbms\_output.put\_line('Klienta numurs: '|| k\_num);

dbms\_output.put\_line('Klienta vārds: '|| k\_var);

dbms\_output.put\_line('Klienta uzvārds: '|| k\_uzv);

dbms\_output.put\_line('Klienta vecums: '|| k\_vecums);

dbms\_output.new\_line;

end IZVADE;

end;

Tabulas KLIENTI izveide:

create table KLIENTI OF TIPS\_KLIENTS

nested table VESTURE\_NOMA STORE AS TAB1;

Tipa TIPS\_FIRMA un tajā ietilpošo objektu tipu izveide.

TIPS\_AUTO\_REF izveide, kas satur atsauci uz automašīnu, kas pieder firmai.

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

REF\_AUTO REF TIPS\_AUTO

);

NOMAS\_MASINAS objekta tipa izveide, kas satur kolekciju at automašīnu atsaucēm ,aks pieder konkrētajai autonomas firmai.

create or replace type NOMAS\_MASINAS as table of TIPS\_AUTO\_REF;

Autonomas firmas adreseses tipa objekta izveide ar tai definētu CONSTRUCOR funkciju:

create or replace type TIPS\_ADRESE as object (

PILSETA varchar2(20),

IELA varchar2(30),

NR\_IELA number,

constructor function TIPS\_ADRESE (PILSETA varchar2, IELA varchar2, NR\_IELA number ) return SELF as result

);

create or replace type body TIPS\_ADRESE as

constructor function TIPS\_ADRESE (PILSETA varchar2, IELA varchar2, NR\_IELA number ) return SELF as result is

BEGIN

SELF.PILSETA:=PILSETA;

SELF.IELA:=IELA;

SELF.NR\_IELA:=NR\_IELA;

RETURN;

END ;

END;

Objektu tipa AUTONOMAS\_FIRMA izveide, ar tia definētu CONSTRUCTOR funkciju, MEMBER procedūru, kas izvada informāciju par autonomas firmu.

create or replace type AUTONOMAS\_FIRMA as object (

F\_NUM number,

F\_NOS varchar2(30),

DIB\_GADS number,

F\_ADRESE TIPS\_ADRESE,

NOMAS\_AUTO NOMAS\_MASINAS,

constructor function AUTONOMAS\_FIRMA (F\_NUM number,F\_NOS varchar2, DIB\_GADS number, F\_ADRESE TIPS\_ADRESE) return SELF as result,

MEMBER procedure F\_IZVADE

);

create or replace type body AUTONOMAS\_FIRMA as

constructor function AUTONOMAS\_FIRMA (F\_NUM number,F\_NOS varchar2, DIB\_GADS number, F\_ADRESE TIPS\_ADRESE) return SELF as result is

begin

SELF.F\_NUM:=F\_NUM;

SELF.F\_NOS:=F\_NOS;

SELF.DIB\_GADS:=DIB\_GADS;

SELF.F\_ADRESE:=F\_ADRESE;

RETURN;

end;

MEMBER procedure F\_IZVADE is

begin

dbms\_output.put\_line('Firmas nr: '|| f\_num);

dbms\_output.put\_line('Firmas nosaukums: '|| f\_nos);

dbms\_output.put\_line('Firmas dibināšanas gads: '|| dib\_gads);

dbms\_output.put\_line('Firmas adrese: '|| f\_adrese.pilseta ||f\_adrese.iela || f\_adrese.nr\_iela);

dbms\_output.new\_line;

end F\_IZVADE;

end;

Tabulas AUTONOMAS\_FIRMAS izveide no tipa AUTONOMAS\_FIRMA ar ligzdtabulu.

create table AUTONOMAS\_FIRMAS of AUTONOMAS\_FIRMA

nested table NOMAS\_AUTO store as TAB\_AUTON;

# Tabulu aizpildīšana un metožu izmantošana

## Constructor function pielietojums aizpildot tabulas ar ierakstiem

### Tabula Automasinas

Tabulā AUTOMASINAS tiek ievietoti 20 ieraksti, katrai autonomas firmai piederošas 5 automašīnas.

begin

insert into AUTOMASINAS values (TIPS\_AUTO(2, 1, MARKA('TOYOTA', 'AURIS', 2012), 14,40000, 25));

insert into AUTOMASINAS values (TIPS\_AUTO(3, 1, MARKA('BMW', 'M4', 2012), 20,45000, 25));

insert into AUTOMASINAS values (TIPS\_AUTO(4, 1, MARKA('BMW', 'TR5', 2010), 29, 65000, 20));

insert into AUTOMASINAS values (TIPS\_AUTO(5, 1, MARKA('BMW', 'M4', 2012), 25,47000, 20));

insert into AUTOMASINAS values (TIPS\_AUTO(6, 2, MARKA('TOYOTA', 'RAV-4', 2018), 20,25000, 40));

insert into AUTOMASINAS values (TIPS\_AUTO(7, 2, MARKA('TOYOTA', 'CHR', 2019), 18, 20000, 40));

insert into AUTOMASINAS values (TIPS\_AUTO(8, 2, MARKA('TOYOTA', 'AURIS', 2012), 30,45000, 20));

insert into AUTOMASINAS values (TIPS\_AUTO(9, 2, MARKA('TOYOTA', 'RAV-4', 2019),4000));

insert into AUTOMASINAS values (TIPS\_AUTO(10, 2, MARKA('TOYOTA', 'CHR', 2020),1000));

insert into AUTOMASINAS values (TIPS\_AUTO(11, 3, MARKA('BMW', 'A1', 2014),30,45000, 25));

insert into AUTOMASINAS values (TIPS\_AUTO(12, 3, MARKA('TOYOTA', 'AURIS', 2012), 30,45000, 20));

insert into AUTOMASINAS values (TIPS\_AUTO(13, 3, MARKA('VOLVO', 'PT3', 2010), 33,48000, 15));

insert into AUTOMASINAS values (TIPS\_AUTO(14, 3, MARKA('VOLVO', 'ART', 2012), 30,45000, 20));

insert into AUTOMASINAS values (TIPS\_AUTO(15, 4, MARKA('SKODA', 'FABIA', 2012),40000));

insert into AUTOMASINAS values (TIPS\_AUTO(16, 4, MARKA('SKODA', 'UTH', 2014), 30,35000, 25));

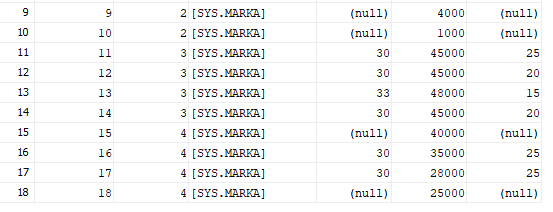
insert into AUTOMASINAS values (TIPS\_AUTO(17, 4, MARKA('FIAT', '3A', 2015), 30,28000, 25));

insert into AUTOMASINAS values (TIPS\_AUTO(18, 4, MARKA('MERCEDES-BENZ', 'S3', 2016),25000));

insert into AUTOMASINAS values (TIPS\_AUTO(19, 3, MARKA('VOLVO', 'KTR', 2014),5000));

insert into AUTOMASINAS values (TIPS\_AUTO(20, 4, MARKA('MERCEDES-BENZ', 'KL3', 2016),14, 35000, 25));

end;

Tā kā ievietojot datus ir izmantotas abas definētās CONSTRUCTOR funkcijas, tad funkcijā neiekļautajiem atribūtiem tiek ieliktas vērtības NULL, kas uzskatāms zemāk redzamajā attēlā.

2.1.1.1.attēls. Ieraksti tabulā AUTOMASINAS.

### Tabula Autonomas\_firmas

Tabulā AUTONOMAS\_FIRMAS tiek ievietoti 5 ieraksti jeb 5 autonomas firmas, izmantojot iepriekšdefinēto CONSTRUCTOR funkciju atbilstošajam tipam. Līdz ar to firmām kolekcijas ar piederošajām automašinām ir ar NULL vērtību.

begin

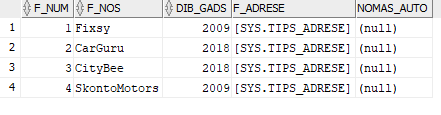
insert into AUTONOMAS\_FIRMAS values (AUTONOMAS\_FIRMA(1, 'Fixsy', 2009, TIPS\_ADRESE('Riga', 'Valnu iela', 12) ));

insert into AUTONOMAS\_FIRMAS values (AUTONOMAS\_FIRMA(2, 'CarGuru', 2018, TIPS\_ADRESE('Riga', 'Slokas iela', 48) ));

insert into AUTONOMAS\_FIRMAS values (AUTONOMAS\_FIRMA(3, 'CityBee', 2018, TIPS\_ADRESE('Riga', 'Balasta iela', 2) ));

insert into AUTONOMAS\_FIRMAS values (AUTONOMAS\_FIRMA(4, 'SkontoMotors', 2009, TIPS\_ADRESE('Valmiera', 'Rigas iela', 24) ));

end;

2.1.2.1.atēls.Ieraksti tabulā AUTONOMAS\_FIRMAS bez datiem kolonā NOMAS\_AUTO

Katrai firmai atbilstošo automašinu atsauču ievietošana tika veikta izmantojot sekojošu vaicājumu, bet katrai firmai izmainot F\_NUM un A\_NUM vērtības.

declare

ats1 ref TIPS\_AUTO;

ats2 ref TIPS\_AUTO;

ats3 ref TIPS\_AUTO;

ats4 ref TIPS\_AUTO;

ats5 ref TIPS\_AUTO;

i number;

k number;

begin

select REF(A) into ats1 from AUTOMASINAS a where a.F\_NUM=1 and a.A\_num=1;

select REF(A) into ats2 from AUTOMASINAS a where a.F\_NUM=1 and a.A\_num=2;

select REF(A) into ats3 from AUTOMASINAS a where a.F\_NUM=1 and a.A\_num=3;

select REF(A) into ats4 from AUTOMASINAS a where a.F\_NUM=1 and a.A\_num=4;

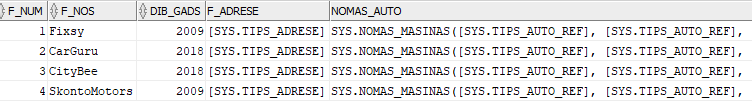
select REF(A) into ats5 from AUTOMASINAS a where a.F\_NUM=1 and a.A\_num=5;

update AUTONOMAS\_FIRMAS b

set b.NOMAS\_AUTO=NOMAS\_MASINAS(TIPS\_AUTO\_REF(ats1), TIPS\_AUTO\_REF(ats2), TIPS\_AUTO\_REF(ats3), TIPS\_AUTO\_REF(ats4), TIPS\_AUTO\_REF(ats5))

where b.F\_NUM=1;

end;



2.1.2.2.attēls. Ieraksti tabulā AUTONOMAS\_FIRMAS ar automašinu atsaucēm

### Tabula KLIENTI

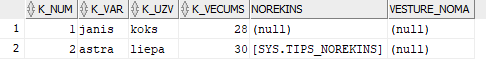
Tabulā KLIENTI tiek ievadīti pirmie divi testa ieraksti izmantojot TIPS\_KLIENTS abus definētos CONSTRUCTOR funkcijas.

begin

insert into KLIENTI values (TIPS\_KLIENTS(1, 'janis', 'koks',28 ));

insert into KLIENTI values (TIPS\_KLIENTS(2, 'astra', 'liepa',30, TIPS\_NOREKINS('SWEDBANK', 'LV80HABA1234') ));

end;



2.1.3.1.attēls. Ierkasti tabulā KLIENTI

Papildus tiek saģenerēti 20 citi ieraksti ar klientu informāciju izmantojot tikai vienu no TIPS\_KLIENTS definētajām constructor funkcijām, atstājot tikai VESTURE\_NOMA kolonnu tukšu, kas aizpildās ar vērtību NULL.

create sequence NUMURI\_2

start with 3

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');

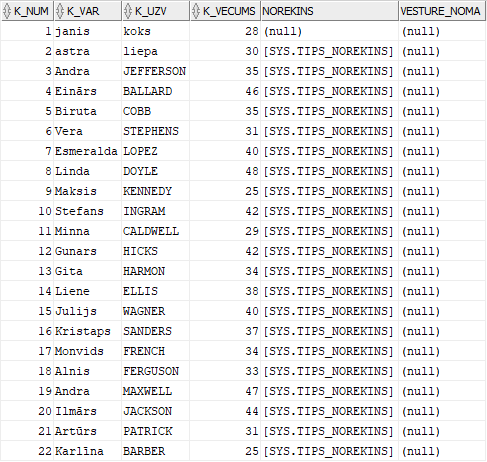
BEGIN

for i in 1..20 loop

insert into KLIENTI values (TIPS\_KLIENTS(NUMURI\_2.NEXTVAL,MASIVS\_VARDI(TRUNC(DBMS\_RANDOM.VALUE(1,120))), MASIVS\_UZVARDS(TRUNC(DBMS\_RANDOM.VALUE(1,500))), TRUNC(DBMS\_RANDOM.VALUE(19,50)), TIPS\_NOREKINS('SWEDBANK', 'LV81HABA1234')));

end loop;

END;

2.1.3.2.attēls. Tabulas KLIENTI ieraksti

### Tabula Rekini

Tabulā REKINI ieraksti tiek automātiski saģenerēti izmantojot zemāk redzamo vaicājumu, atstājot kolonnas CENA, GALA\_SUMMA, REF\_AUTO tukšas, kas tiek aizpildītas pec tam.

create sequence rekini\_nr

start with 1

increment by 1

order;

declare

i number;

BEGIN

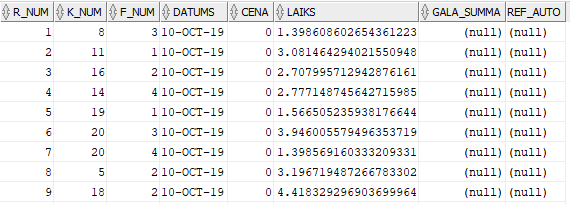
for i in 1..40 loop

insert into REKINI values (TIPS\_REKINS(REKINI\_NR.NEXTVAL, TRUNC(DBMS\_RANDOM.VALUE(1,21)),

TRUNC(DBMS\_RANDOM.VALUE(1,5)),'10-oct-2019',0,DBMS\_RANDOM.VALUE(1,5) ));

end loop;

END;

2.1.4.1.attēls. Tabulas REKINI daļēji aizpildīti ieraksti

Katram rēķinam tiek piesaistīta viena TIPS\_AUTO atsauce, pēc kuras arī tiek ievietota attiecīgās automašinas nomas cena kolonnā CENA.

declare

i number;

k number;

ats1 ref TIPS\_AUTO;

BEGIN

for i in 1..40 loop

k:=TRUNC(DBMS\_RANDOM.VALUE(1,21));

select ref(a) into ats1 from AUTOMASINAS a where a.A\_NUM=K;

update REKINI b

set b.REF\_AUTO=ats1

where b.R\_NUM=i;

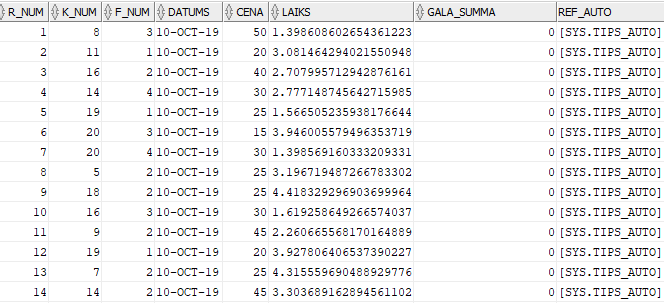
update REKINI b

set b.CENA=DEREF(value(b).REF\_AUTO).NOMAS\_CENA

where b.R\_NUM=i;

end loop;

END;



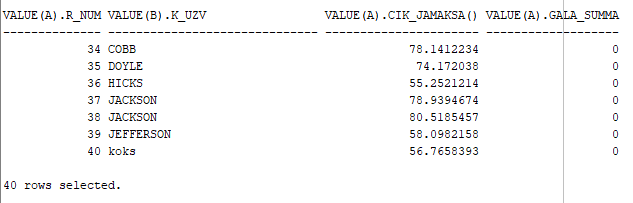
2.1.4.2.attēls. Ieraksti tabulā REKINI.

## Member procedure un function pielietojums

Ar TIPS\_REKINS definētās MEMBER funkcijas palīdzību tiek aprēķināta gala summa.

select VALUE(A).R\_NUM, VALUE(B).K\_UZV, VALUE(A).CIK\_JAMAKSA(), VALUE(A).GALA\_SUMMA

from REKINI A, KLIENTI B

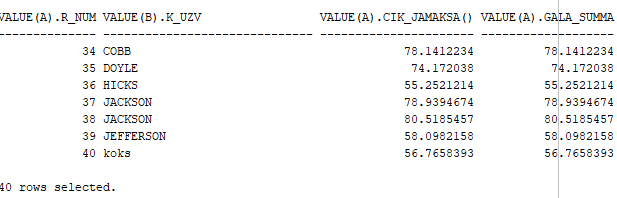
where A.K\_NUM=B.K\_NUM;

2.2.1.attēls. Katra rēķina gala summa

Šīs gala summas vērtība arī tiek ievietota tabulā REKINI kolonnā GALA\_SUMMA.

update REKINI a

set GALA\_SUMMA=VALUE(A).CIK\_JAMAKSA();

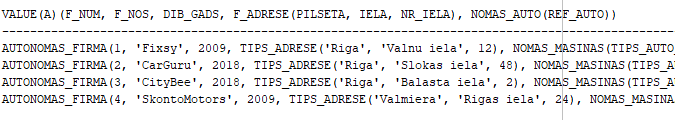


2.2.2.attēls. Tabulas REKINI infomrācijua par GALA\_SUMMA

Vaicājums, kas atgriež informāciju par katru autonomas firmu.

select VALUE(A)

from AUTONOMAS\_FIRMAS A;



2.2.3.attēls. Vaicājuma rezultāts

Tipam AUTONOMAS\_FIRMA ir arī definēta MEMBER procedūra F\_IZVADE, kas atgriež informāciju par pašu firmu pārskatāmāka veidā.

set serveroutput on size 5000;

declare

PIEMERS\_1 AUTONOMAS\_FIRMA;

begin

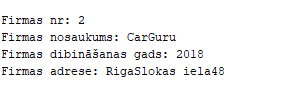
select value(a) into PIEMERS\_1

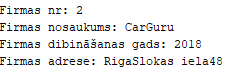
from AUTONOMAS\_FIRMAS A

where value(a).F\_NUM=2;

PIEMERS\_1.F\_IZVADE;

end;



2.2.4.attēls.Procedūras F\_IZVADE rezultāts

Līdzīga izvades procedūra ir definēta arī TIPS\_KLIENTS.

set serveroutput on size 5000;

declare

PIEMERS\_2 TIPS\_KLIENTS;

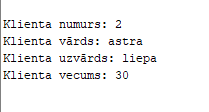
begin

select value(a) into PIEMERS\_2

from KLIENTI A

where value(a).K\_NUM=2;

PIEMERS\_2.IZVADE;

end;

2.2.5.attēls. Procedūras IZVADE rezultāts

## ORDER funkcijas pielietojums

Tipam TIPS\_AUTO ir definēta ORDER member funkcija, kas salīdzina automašinas noams cenu, un atgriež vērtību –1, ja tā ir zemāka par izvēlēto, 0, ja sakrīt un 1 ja tā ir augstāka.

Atbilstošais vaicājums:

select VALUE(A).A\_NUM AS NUM, VALUE(A).A\_MARKA.A\_MARKA AS MARKAS,

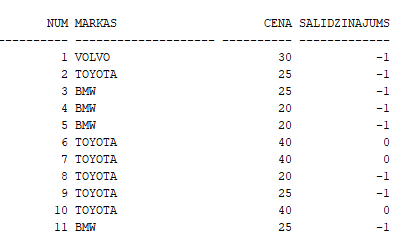
VALUE(A).NOMAS\_CENA AS CENA, VALUE(A).SALIDZINAT\_NOMAS\_CENU(C.OBJ) AS SALIDZINAJUMS

FROM AUTOMASINAS A,

(select VALUE(B) as OBJ

from AUTOMASINAS B

where VALUE(B).A\_NUM=6)C;



2.3.1.attēls. Vaicājuma rezultāts iesaistot ORDER funkciju

Nedaudz pārmainot šo vaicājumu, var iegūt sarakstu ar atomašīnām, kuru nomas cena ir zemāka par izvēlēto.

Atbilsotošais vaicājums;:

select VALUE(A).A\_NUM AS NUM, VALUE(A).A\_MARKA.A\_MARKA AS MARKAS,

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

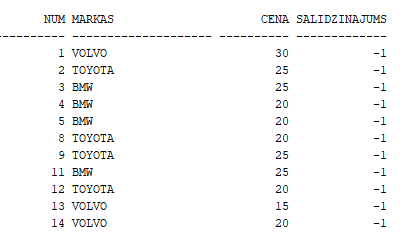
FROM AUTOMASINAS A,

(select VALUE(B) as OBJ

from AUTOMASINAS B

where VALUE(B).A\_NUM=6)C

**where VALUE(A).SALIDZINAT\_NOMAS\_CENU(C.OBJ)=-1;**



2.3.2.attēls.Vaicājuma reuzltāts ar ORDER funkcijas iesaisti

## MAP funkcijas pielietojums

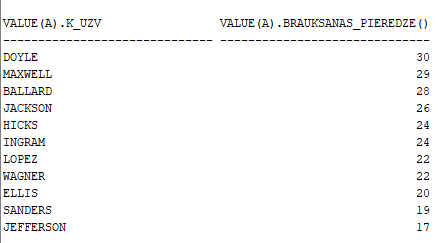
Objektu tipam TIPS\_KLIENTS ir definēta MAP member funkcija, kura atgriež vērtību par katra klienta brauksāsnas pieredzi jeb stāžu.

Attiecīgais vaicājums, kur klienti ir sakārtoti disltošā secībā pēc to braukšanas pieredzes:

select value(a).k\_uzv, value(a).BRAUKSANAS\_PIEREDZE()

from KLIENTI A

order by value(a).BRAUKSANAS\_PIEREDZE() desc;



2.4.1.attēls. Vaicājuma rezultāts ar MAP funkcijas BRAUKSANAS\_PIEREDZE iesaistīšanu

# Objekta skata izveide, kas ietver MEMBER metodi

Lai samazinātu atribūtu skaitu katrā no tālāk definētajiem skatiem , tiek nodefinēti jauni objektu tipi ar tajos ietilpstošajiem atribūtiem.

Objekta tips T\_AUTO uz kura balstīsies objektu skats SKATS\_AUTO.

create type t\_auto as object (

a\_num number,

f\_num number,

a\_marka varchar2(20),

modelis varchar2(20)

);

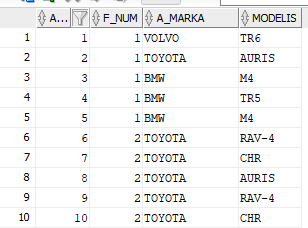
Objekta skata SKATS\_AUTO definēšana:

create view SKATS\_AUTO of T\_AUTO

with object identifier (A\_NUM) as

select a.a\_num, a.f\_num, a.a\_marka.a\_marka, a.a\_marka.modelis

from automasinas a;



3.1.attēls. Ieraksti objektu skata SKATS\_AUTO tabulā

Objekta tipa T\_KLIENTS definēšana ar MEMBER tipa funkciju BRAUKSANAS\_PIEREDZE, uz kura balstīsies objektu skats SKATS\_KLIENTS.

create or replace type t\_klients as object (

k\_num number,

k\_var varchar2(25),

k\_uzv varchar2(30),

k\_vecums number,

MEMBER function BRAUKSANAS\_PIEREDZE return number

);

create or replace type body t\_klients as

MEMBER function BRAUKSANAS\_PIEREDZE return number is

begin

return self.k\_vecums-18;

end BRAUKSANAS\_PIEREDZE;

end;

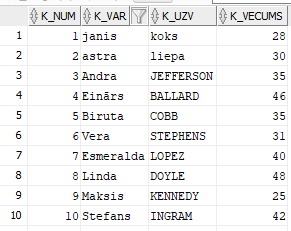
Objekta skata SKATS\_KLIENTS definēšana.

Create or replace view SKATS\_KLIENTS of T\_KLIENTS

with object identifier (K\_NUM) as

select a.K\_NUM, a.K\_VAR, a.K\_UZV, a.K\_VECUMS

from KLIENTI a;



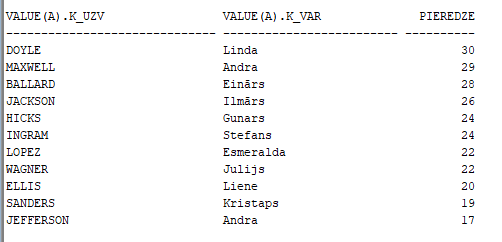
3.2.attēls. Ieraksti ojektu skata SKATS\_KLIENTS tabulā

Vaicājums objekta skatam SKATS\_KLIENTS, izmantojot MEMBER funkciju BRAUKSANAS\_PIEREDZE:

select VALUE(A).K\_UZV, VALUE(A).K\_VAR, VALUE(A).BRAUKSANAS\_PIEREDZE() as PIEREDZE

from SKATS\_KLIENTS A

order by VALUE(A).BRAUKSANAS\_PIEREDZE() DESC;

3.3.attēls. Objekta skata vaicājums izmantojot MEMBER funkciju BRAUKSANAS\_PIEREDZE

Objektu tipa T\_REKINS izveide, kas sevī ietver atsauces uz tipiem T\_AUTO un T\_KLIENTS.

create or replace type T\_REKINS as object (

r\_num number,

k\_num number,

ATSAUCE\_KLIENTS REF T\_KLIENTS,

f\_num number,

cena number,

laiks real,

gala\_summa real,

ATSAUCE\_AUTO REF T\_AUTO

);

Objektu skata definēšana, kas balstās uz T\_REKINS objekta tipu, kā arī sasaista objketu skatus SKATS\_KLIENTS un SKATS\_AUTO.

create or replace view SKATS\_REKINS of T\_REKINS

with object identifier (R\_NUM) as

select A.R\_NUM, A.K\_NUM, MAKE\_REF(SKATS\_KLIENTS, A.K\_NUM) ATSAUCE\_KLIENTS,

A.F\_NUM, A.CENA, A.LAIKS, A.GALA\_SUMMA,

MAKE\_REF(SKATS\_AUTO, A.F\_NUM) ATSAUCE\_AUTO

from REKINI A;

T\_AUTO

SKATS\_REKINS

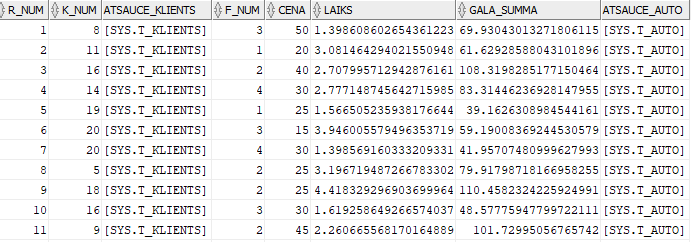
T\_KLIENTS

T\_REKINS

SKATS\_AUTO

SKATS\_KLIENTS

3.4.attēls. Shēma, kas attaino objektu skatu sasaisti

3.5.attēls. Ieraksti objketa skata SKATS\_REKINS ieraksti

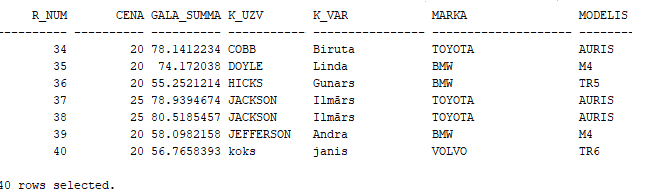
Vaicājums izmantojot objektu skatu SKATS\_REKINS:

select VALUE(A).R\_NUM as R\_NUM, VALUE(A).CENA as CENA, VALUE(A).GALA\_SUMMA as GALA\_SUMMA,

DEREF(VALUE(A).ATSAUCE\_KLIENTS).K\_UZV as K\_UZV, DEREF(VALUE(A).ATSAUCE\_KLIENTS).K\_VAR as K\_VAR,

DEREF(VALUE(A).ATSAUCE\_AUTO).A\_MARKA as MARKA, DEREF(VALUE(A).ATSAUCE\_AUTO).MODELIS as MODELIS

from SKATS\_REKINS A;



3.6.attēls. Objektu skata SKATS\_REKINS vaicājuma rezultāts

Vaicājums izmantojot objektuu skatu SKATS\_REKINS un tā pārmantoto MEMBER funkciju BRAUKSANAS\_PIEREDZE no objektu skata SKATS\_KLIENTS.

select VALUE(A).R\_NUM as R\_NUM, VALUE(A).CENA as CENA, VALUE(A).GALA\_SUMMA as GALA\_SUMMA,

DEREF(VALUE(A).ATSAUCE\_KLIENTS).K\_UZV as K\_UZV, DEREF(VALUE(A).ATSAUCE\_KLIENTS).K\_VAR as K\_VAR,

DEREF(VALUE(A).ATSAUCE\_KLIENTS).BRAUKSANAS\_PIEREDZE() as PIEREDZE, DEREF(VALUE(A).ATSAUCE\_AUTO).A\_MARKA as MARKA, DEREF(VALUE(A).ATSAUCE\_AUTO).MODELIS as MODELIS

from SKATS\_REKINS A;



3.7.attēls. Vaicājuma rezultāti.

# Secinājumi

Objektu metodes jeb subprogrammas, kas tiek deklarētas objektu tipu definīcijā, ļauj sekmēt atbilstošo objektu uzvedību un iespējas to manipulācijai. Šīs metodes ļauj piekļūt datiem, kas ir katrā objektā. Ar šo metožu palīdzību var veikt aprēķinus dažādas objektu salīdzināšanas un kārtošanas operācijas, kā arī izvades procedūrass izmantojot universālo MEMBER tipa metodi.

Šajā darbā definētās objektu metodes tikai daļēji atspulguļo to iespējas un pielietojumus, un tai skaitā diezgan vienkāršos veidos. Tika relizēta MAP meber funkcija, kas aprēķina klientu braukšanas pieredzi. ORDER funkcija, kas salīdzina nomas automašinu cenas likmi pret izvēlēto nomas auto cenu, un atgriež vērtības balstoties uz to vai ir lielāka, mazāka vai vienāda, kuru pēc tam var izmantot, lai atlasītu vēlamās nomas mašīnas. Tika realizētas arī MEMBER univeršlā tipa metodes. Viena MEMBER metode atgrieza summu par auto nomu, balstoties uz cenu un nomas laiku. Tiak izveidotas arī divas MEMBER procedūras, kas veic informācijas izvadi.

Visvairāk tika definētas CONSTRUCTOR tipa funkcijas, lai izveidotu objektus. Ar šīs metodes palīdzību var izveidot objektus ar mainīgu atribūtu daudzumu, un kas automātiski ievieto vērtību NULL tajos atribūtos, kuri nav iekļauti attiecīgajā CONSTRUCTOR funkcijā. Šīs NULL vērtības pēc tam ir vienkārši aizstāt ar nepieciešamajiem datiem izmantot komandu UPDATE.

Lai kaut cik objektīvi spriestu par metožu izmantošanas iespējām, būtu vajadzīgs realizēt vairākas instances katrai tipa metodei dažādās variācijās. Bet no pielietotajām metodēm šī darba ietvaros, var secināt, ka universālā MEMBER tipa metodei ir visplašākās pielietošanas iespējas, balstoties vai tā tiek definēta kā PROCEDURE vai FUNCTION.

Darba ietvaros tiak arī izveidoti objektu skati SKATS\_AUTO, SKATS\_KLIENTS un SKATS\_REKINS. Šajāa gadījumā tika arī izveidota struktūra, kurā objektu skats ietver sevī citus objektu skatus. Bet šī farba ietvaros netika aapskatītas priekšrocības objektu skatu veidošanai.

Šī darba ietveros fokuss bija uz objektu tipu, objektu metožu un objektu skatu izveidi, izveidošanas prasmes apguvi, līdz ar to testa datu skiats bija stipri mazs, bet šīs lietas jau tika apgūtas iepriekšējo divu praktisko darbu ietveros.

Kopsummā, 3.praktiskais darbs ir izstrādāts labi; ir viekti visi darba nostādnē uzskaitītie uzdevumi.