

**Проект**

**По**

**„Програмиране на бази данни“**

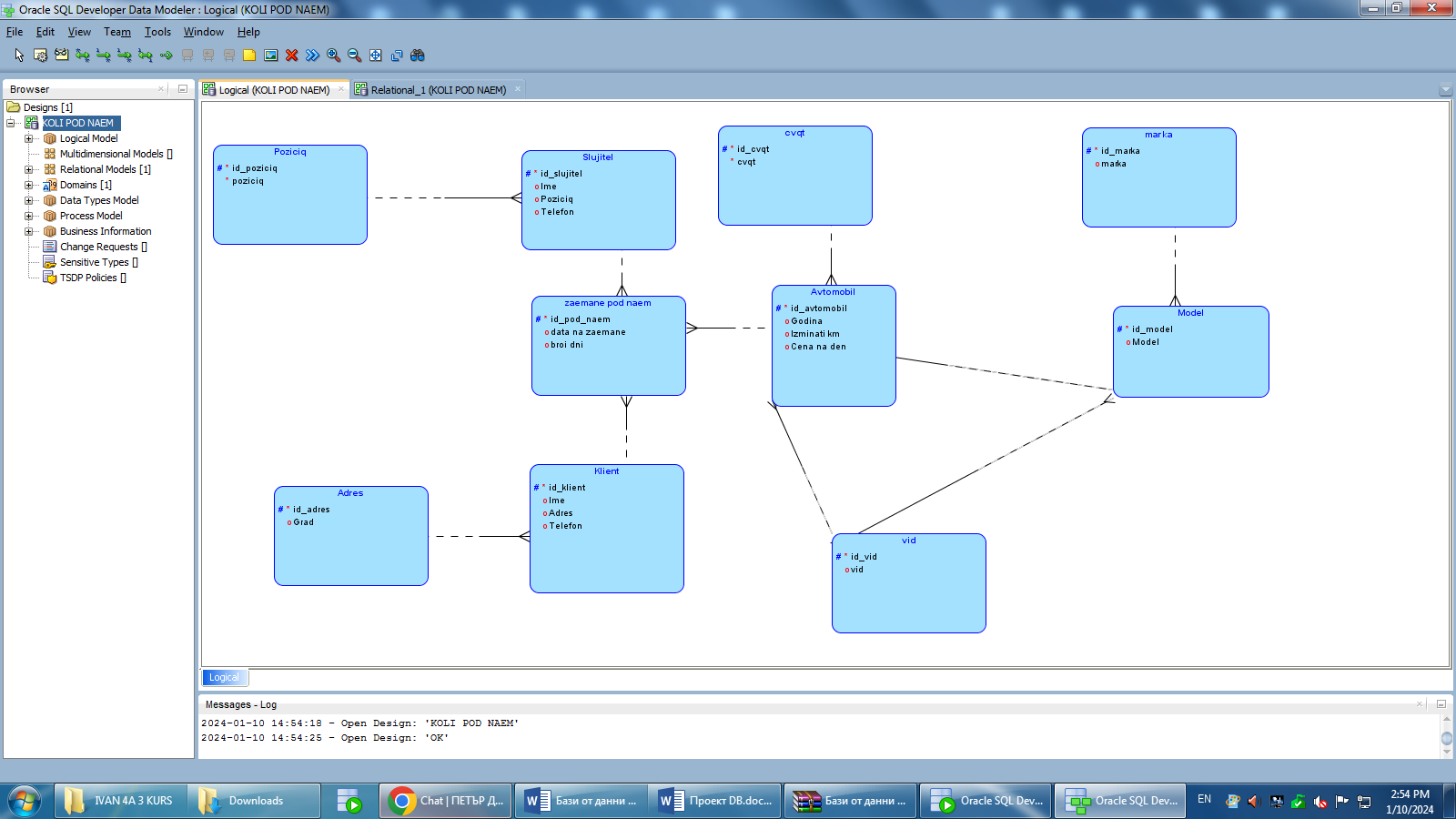
Студент: Иван Пламенов Колев ФН:21621580 гр.4 III к. СИТ

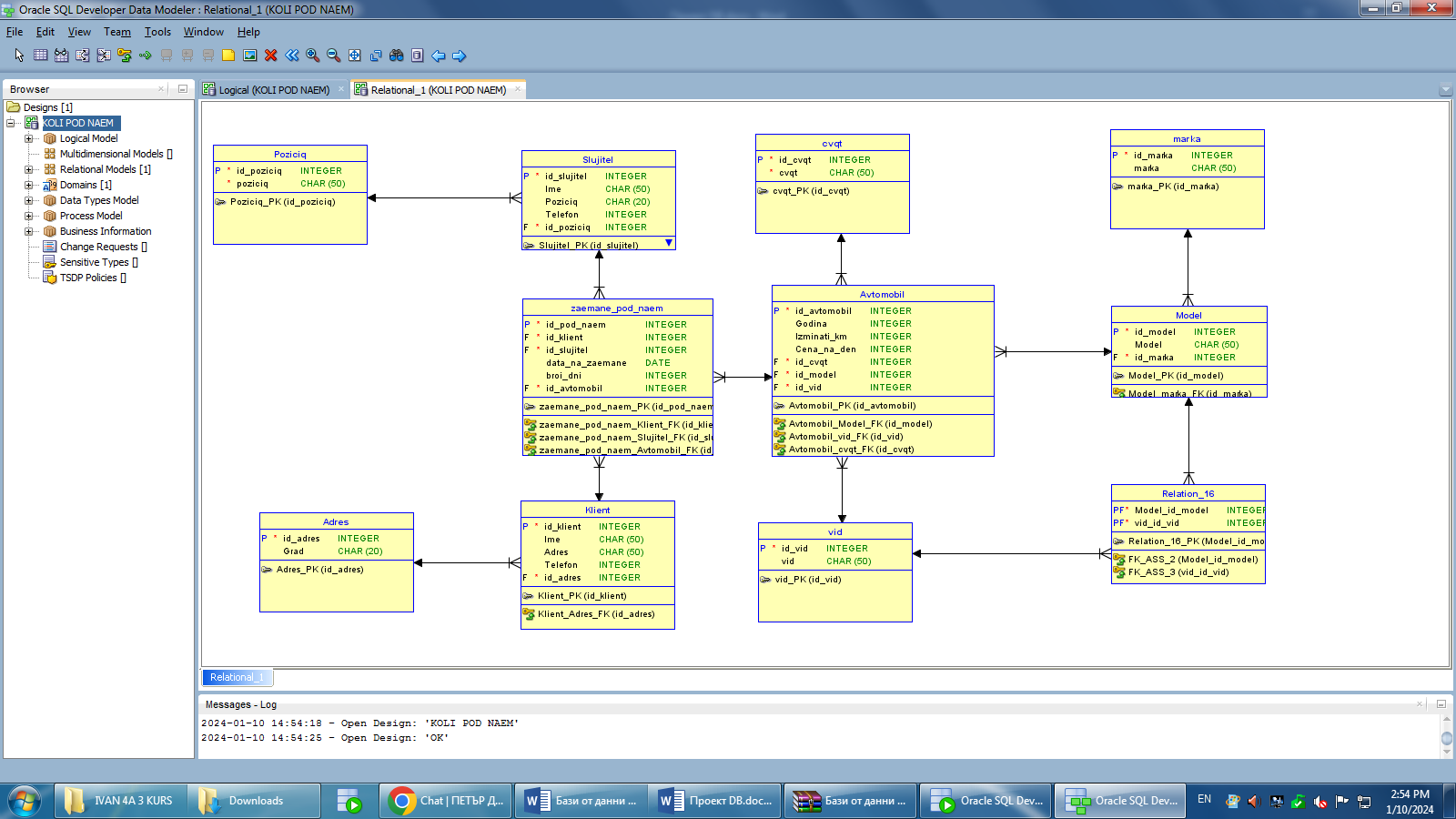
Описание

Проектът представлява приложение за управление на сайт за коли под наем с различни функции.

1. **Заявка за налични автомобили:** Потребителите могат да пускат заявка за наличен автомобил, като сложат филтри, като дата, час и др.
2. **Резервация на автомобил:** Избор на автомобил по марка, модел и наличност в избраните дати.
3. **Редакция на резервация:** Промяна на вече направена резервация, като промяна на автомобил, дата, час и други детайли по резервацията, ако е възможно.
4. **Изтриване на резервация:** Потребителите могат да изтриват вече направени резервации от тях.
5. **Справки:** Последни 10 отдавания, наети автомобили от клиент, отдадени автомобили от служител.

**2.Модели на БД с пояснения за таблиците**





1. Zaemane pod name:

* Id\_pod\_naem: уникален идентификатор за отдаване под наем.
* Data\_na\_zaemane: датата на заемане на автомобил.
* Broi\_dni: период от време, в който даден автомобил ще бъде отдаден под наем.

1. Slujitel:

* Id\_slujitel: уникален идентификатор на служителя.
* Ime: име на служител.
* Poziciq:позицията, която заема даден служител.
* Telefon: телефонен номер на служител.

1. Poziciq:

* Id\_poziciq: уникален идентификатор на позицията.
* Poziciq: позицията на служител във фирмата.

1. Klient:

* Id\_klient: уникален идентификатор на клиент.
* Ime: име на клиент.
* Adres: Адрес на клиент.
* Telefon: телефонен номер на клиент.

1. Adres:

* Id\_adres: уникален идентификатор за адреса на клиент.
* Grad: градът, в който живее клиента.

1. Аvtomobili:

* Id\_avtomobili: уникален идентификатор на автомобила.
* Godina: година на производство на автомобила.
* Izminati\_kilometri: изминати километри на автомобила от датата на производство.
* Cena\_na\_den: Цена на автомобила за отдаване под наем за един ден.

1. Cvqt:

* Id\_cvqt: уникален идентификатор за цветовете на автомобилите.
* Cvqt: цвят на автомобила.

1. Vid:

* Id\_vid: уникален идентификатор за вида на автомобила.
* Vid: вид на автомобила(седан, купе, комби и тн.).

1. Model:

* Id\_model:уникален идентификатор за модела на автомобила.
* Model:Модел на автомобила, като допълнение на марката.

10.Marka:

* Id\_marka:уникален идентификатор за марката на автомобила.
* Marka: марка на автомобила(ауди, мерцедес, бмв).

11.Relation 16: връзката между марка и вид на автомобила.

**3. Кратко описание на избраните средства за реализация на пркекта**

Избрах HTML, CSS, PHP, MySQL за създаването на това приложение. HTML и CSS изиграха главна роля в създаването на потребителския интерфейс. PHP помогна за изработването на функционалностите а MySQL помогна за съхраняването на информацията.

**4. Код на базата данни**

CREATE TABLE Adres

( id\_adres INTEGER NOT NULL , Grad CHAR (20)

) ;

ALTER TABLE Adres ADD CONSTRAINT Adres\_PK PRIMARY KEY ( id\_adres ) ;

CREATE TABLE Avtomobil

(

id\_avtomobil INTEGER NOT NULL ,

Godina INTEGER ,

Izminati\_km INTEGER ,

Cena\_na\_den INTEGER ,

id\_cvqt INTEGER NOT NULL ,

id\_model INTEGER NOT NULL ,

id\_vid INTEGER NOT NULL

) ;

ALTER TABLE Avtomobil ADD CONSTRAINT Avtomobil\_PK PRIMARY KEY ( id\_avtomobil ) ;

CREATE TABLE Klient

(

id\_klient INTEGER NOT NULL ,

Ime CHAR (50) ,

Adres CHAR (50) ,

Telefon INTEGER ,

id\_adres INTEGER NOT NULL

) ;

ALTER TABLE Klient ADD CONSTRAINT Klient\_PK PRIMARY KEY ( id\_klient ) ;

CREATE TABLE Model

(

id\_model INTEGER NOT NULL ,

Model CHAR (50) ,

id\_marka INTEGER NOT NULL

) ;

ALTER TABLE Model ADD CONSTRAINT Model\_PK PRIMARY KEY ( id\_model ) ;

CREATE TABLE Poziciq

(

id\_poziciq INTEGER NOT NULL ,

poziciq CHAR (50) NOT NULL

) ;

ALTER TABLE Poziciq ADD CONSTRAINT Poziciq\_PK PRIMARY KEY ( id\_poziciq ) ;

CREATE TABLE Relation\_16

(

Model\_id\_model INTEGER NOT NULL ,

vid\_id\_vid INTEGER NOT NULL

) ;

ALTER TABLE Relation\_16 ADD CONSTRAINT Relation\_16\_PK PRIMARY KEY ( Model\_id\_model, vid\_id\_vid ) ;

CREATE TABLE Slujitel

(

id\_slujitel INTEGER NOT NULL ,

Ime CHAR (50) ,

Poziciq CHAR (20) ,

Telefon INTEGER ,

id\_poziciq INTEGER NOT NULL

) ;

ALTER TABLE Slujitel ADD CONSTRAINT Slujitel\_PK PRIMARY KEY ( id\_slujitel ) ;

CREATE TABLE cvqt

( id\_cvqt INTEGER NOT NULL , cvqt CHAR (50) NOT NULL

) ;

ALTER TABLE cvqt ADD CONSTRAINT cvqt\_PK PRIMARY KEY ( id\_cvqt ) ;

CREATE TABLE marka

( id\_marka INTEGER NOT NULL , marka CHAR (50)

) ;

ALTER TABLE marka ADD CONSTRAINT marka\_PK PRIMARY KEY ( id\_marka ) ;

CREATE TABLE vid

( id\_vid INTEGER NOT NULL , vid CHAR (50)

) ;

ALTER TABLE vid ADD CONSTRAINT vid\_PK PRIMARY KEY ( id\_vid ) ;

CREATE TABLE zaemane\_pod\_naem

(

id\_pod\_naem INTEGER NOT NULL ,

id\_klient INTEGER NOT NULL ,

id\_slujitel INTEGER NOT NULL ,

data\_na\_zaemane DATE ,

broi\_dni INTEGER ,

id\_avtomobil INTEGER NOT NULL

) ;

**3.Създаване на връзки между таблиците**

ALTER TABLE zaemane\_pod\_naem ADD CONSTRAINT zaemane\_pod\_naem\_PK PRIMARY KEY ( id\_pod\_naem ) ;

ALTER TABLE Avtomobil ADD CONSTRAINT Avtomobil\_Model\_FK FOREIGN KEY ( id\_model ) REFERENCES Model ( id\_model ) ;

ALTER TABLE Avtomobil ADD CONSTRAINT Avtomobil\_cvqt\_FK FOREIGN KEY ( id\_cvqt ) REFERENCES cvqt ( id\_cvqt ) ;

ALTER TABLE Avtomobil ADD CONSTRAINT Avtomobil\_vid\_FK FOREIGN KEY ( id\_vid ) REFERENCES vid ( id\_vid ) ;

ALTER TABLE Relation\_16 ADD CONSTRAINT FK\_ASS\_2 FOREIGN KEY ( Model\_id\_model ) REFERENCES Model ( id\_model ) ;

ALTER TABLE Relation\_16 ADD CONSTRAINT FK\_ASS\_3 FOREIGN KEY ( vid\_id\_vid ) REFERENCES vid ( id\_vid ) ;

ALTER TABLE Klient ADD CONSTRAINT Klient\_Adres\_FK FOREIGN KEY ( id\_adres ) REFERENCES Adres ( id\_adres ) ;

ALTER TABLE Model ADD CONSTRAINT Model\_marka\_FK FOREIGN KEY ( id\_marka ) REFERENCES marka ( id\_marka ) ;

ALTER TABLE Slujitel ADD CONSTRAINT Slujitel\_Poziciq\_FK FOREIGN KEY ( id\_poziciq ) REFERENCES Poziciq ( id\_poziciq ) ;

ALTER TABLE zaemane\_pod\_naem ADD CONSTRAINT zaemane\_pod\_naem\_Avtomobil\_FK FOREIGN KEY ( id\_avtomobil ) REFERENCES Avtomobil ( id\_avtomobil ) ;

ALTER TABLE zaemane\_pod\_naem ADD CONSTRAINT zaemane\_pod\_naem\_Klient\_FK FOREIGN KEY ( id\_klient ) REFERENCES Klient ( id\_klient ) ;

ALTER TABLE zaemane\_pod\_naem ADD CONSTRAINT zaemane\_pod\_naem\_Slujitel\_FK FOREIGN KEY ( id\_slujitel ) REFERENCES Slujitel ( id\_slujitel ) ;

**4.Процедури**

CREATE OR REPLACE PROCEDURE "ADR\_INS"

(v\_GRAD ADRES.grad%TYPE)

AS

BEGIN

INSERT INTO ADRES(grad)

Values (v\_grad);

END;

--------------------------------------------------------

-- DDL for Procedure ADR\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "ADR\_UPD"

(v\_id\_adres ADRES.id\_adres%TYPE,

v\_grad ADRES.grad%TYPE)

AS

BEGIN

UPDATE ADRES set grad= v\_grad

where id\_adres= v\_id\_adres;

END;

--------------------------------------------------------

-- DDL for Procedure AVT\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "AVT\_INS"

(v\_godina AVTOMOBIL.godina%TYPE,

v\_izminati\_km AVTOMOBIL.izminati\_km%TYPE,

v\_cena\_na\_den AVTOMOBIL.cena\_na\_den%TYPE,

v\_CVQT AVTOMOBIL.ID\_CVQT%TYPE,

v\_MODEL AVTOMOBIL.ID\_MODEL%TYPE,

v\_VID AVTOMOBIL.ID\_VID%TYPE)

AS

BEGIN

INSERT INTO AVTOMOBIL(godina, izminati\_km, cena\_na\_den, ID\_CVQT, ID\_MODEL, ID\_VID)

Values (v\_godina, v\_izminati\_km,v\_cena\_na\_den, v\_CVQT, v\_MODEL, v\_VID);

END;

--------------------------------------------------------

-- DDL for Procedure AVT\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "AVT\_UPD"

(v\_id\_avtomobil AVTOMOBIL.id\_avtomobil%TYPE,

v\_godina AVTOMOBIL.godina%TYPE,

v\_izminati\_km AVTOMOBIL.izminati\_km%TYPE,

v\_cena\_na\_den AVTOMOBIL.cena\_na\_den%TYPE,

v\_CVQT AVTOMOBIL.ID\_CVQT%TYPE,

v\_MODEL AVTOMOBIL.ID\_MODEL%TYPE,

v\_VID AVTOMOBIL.ID\_VID%TYPE)

AS

BEGIN

UPDATE AVTOMOBIL set godina= v\_godina,

izminati\_km= v\_izminati\_km,

cena\_na\_den= v\_cena\_na\_den,

id\_cvqt= v\_CVQT,

id\_model= v\_model,

id\_vid= v\_VID

where id\_avtomobil= v\_id\_avtomobil;

END;

--------------------------------------------------------

-- DDL for Procedure CVQT\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "CVQT\_INS"

(v\_cvqt CVQT.cvqt%TYPE)

AS

BEGIN

INSERT INTO CVQT(cvqt)

Values (v\_cvqt);

END;

--------------------------------------------------------

-- DDL for Procedure CVQT\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "CVQT\_UPD"

(v\_id\_cvqt CVQT.id\_cvqt%TYPE,

v\_cvqt CVQT.cvqt%TYPE)

AS

BEGIN

UPDATE CVQT set cvqt= v\_cvqt

where id\_cvqt= v\_id\_cvqt;

END;

--------------------------------------------------------

-- DDL for Procedure FIND\_EMPLOYEE

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "FIND\_EMPLOYEE" (KEYWORDS VARCHAR2) IS

KEYWORDS\_new VARCHAR2(500);

BEGIN

KEYWORDS\_new := '%' || KEYWORDS || '%';

DBMS\_OUTPUT.PUT\_LINE('Employees whose name contains ' || KEYWORDS);

FOR EMP\_RECORD IN (

SELECT s.ime AS slujitel\_ime, p.poziciq

FROM slujitel s

JOIN poziciq p ON s.id\_poziciq = p.id\_poziciq

WHERE s.ime LIKE KEYWORDS\_new

)

LOOP

DBMS\_OUTPUT.PUT\_LINE(EMP\_RECORD.slujitel\_ime || ' ' || EMP\_RECORD.poziciq);

END LOOP;

END FIND\_EMPLOYEE;

--------------------------------------------------------

-- DDL for Procedure KLNT\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "KLNT\_INS"

(v\_ime KLIENT.ime%TYPE,

v\_ULICA KLIENT.ULICA%TYPE,

v\_telefon KLIENT.telefon%TYPE,

v\_id\_adres KLIENT.ID\_ADRES%TYPE)

AS

BEGIN

INSERT INTO KLIENT(ime, ULICA, telefon, ID\_ADRES)

Values (v\_ime, v\_ULICA, v\_telefon, v\_id\_adres);

END;

--------------------------------------------------------

-- DDL for Procedure MDL\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "MDL\_INS"

(v\_model MODELL.model%TYPE,

v\_MARKA MODELL.ID\_MARKA%TYPE)

AS

BEGIN

INSERT INTO MODELL(model, ID\_MARKA)

Values (v\_model, v\_MARKA);

END;

--------------------------------------------------------

-- DDL for Procedure MDL\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "MDL\_UPD"

(v\_id\_model MODELL.id\_model%TYPE,

v\_model MODELL.model%TYPE)

AS

BEGIN

UPDATE MODELL set model= v\_model

where id\_model= v\_id\_model;

END;

--------------------------------------------------------

-- DDL for Procedure MODEL\_TYPE\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "MODEL\_TYPE\_INS"

(v\_model\_id\_model Model\_Type.model\_id\_model%TYPE,

v\_vid\_id\_vid Model\_Type.vid\_id\_vid%TYPE)

AS

BEGIN

INSERT INTO Model\_Type(model\_id\_model, vid\_id\_vid)

Values (v\_model\_id\_model, v\_vid\_id\_vid);

END;

--------------------------------------------------------

-- DDL for Procedure MRK\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "MRK\_INS"

(v\_marka MARKA.marka%TYPE)

AS

BEGIN

INSERT INTO MARKA(marka)

Values (v\_marka);

END;

--------------------------------------------------------

-- DDL for Procedure MRK\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "MRK\_UPD"

(v\_id\_marka MARKA.id\_marka%TYPE,

v\_marka MARKA.marka%TYPE)

AS

BEGIN

UPDATE MARKA set marka= v\_marka

where id\_marka= v\_id\_marka;

END;

--------------------------------------------------------

-- DDL for Procedure POS\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "POS\_INS"

(v\_poziciq POZICIQ.poziciq%TYPE,

v\_zaplata POZICIQ.zaplata%TYPE)

AS

BEGIN

INSERT INTO POZICIQ(poziciq, zaplata)

Values (v\_poziciq, v\_zaplata);

END;

--------------------------------------------------------

-- DDL for Procedure POS\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "POS\_UPD"

(v\_id\_poziciq POZICIQ.poziciq%TYPE,

v\_poziciq POZICIQ.poziciq%TYPE,

v\_zaplata POZICIQ.zaplata%TYPE)

AS

BEGIN

UPDATE POZICIQ set poziciq= v\_poziciq,

zaplata= v\_zaplata

where id\_poziciq= v\_id\_poziciq;

END;

--------------------------------------------------------

-- DDL for Procedure SLUJ\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "SLUJ\_INS"

(v\_ime SLUJITEL.ime%TYPE,

v\_poziciq SLUJITEL.poziciq%TYPE,

v\_telefon SLUJITEL.telefon%TYPE,

v\_id\_poziciq SLUJITEL.ID\_POZICIQ%TYPE)

AS

BEGIN

INSERT INTO SLUJITEL(ime, poziciq, telefon, ID\_POZICIQ)

Values (v\_ime, v\_poziciq, v\_telefon, v\_id\_poziciq);

END;

--------------------------------------------------------

-- DDL for Procedure SLUJ\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "SLUJ\_UPD"

(v\_id\_slujitel SLUJITEL.id\_slujitel%TYPE,

v\_ime SLUJITEL.ime%TYPE,

v\_poziciq SLUJITEL.poziciq%TYPE,

v\_telefon SLUJITEL.telefon%TYPE,

v\_id\_poziciq SLUJITEL.ID\_POZICIQ%TYPE)

AS

BEGIN

UPDATE SLUJITEL set ime= v\_ime,

poziciq= v\_poziciq,

telefon= v\_telefon,

id\_poziciq= v\_id\_poziciq

where id\_slujitel= v\_id\_slujitel;

END;

--------------------------------------------------------

-- DDL for Procedure SPR\_EMPL\_POS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "SPR\_EMPL\_POS" (V\_POS\_NAME slujitel.poziciq%TYPE) IS

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Employees with ' || V\_POS\_NAME || ' position');

FOR EMP\_RECORD IN (

SELECT s.ime AS slujitel\_ime

FROM slujitel s

JOIN poziciq p ON s.id\_poziciq = p.id\_poziciq

WHERE p.poziciq = V\_POS\_NAME

)

LOOP

DBMS\_OUTPUT.PUT\_LINE(EMP\_RECORD.slujitel\_ime);

END LOOP;

END spr\_Empl\_pos;

--------------------------------------------------------

-- DDL for Procedure VID\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "VID\_INS"

(v\_vid VID.vid%TYPE)

AS

BEGIN

INSERT INTO VID(vid)

Values (v\_vid);

END;

--------------------------------------------------------

-- DDL for Procedure VID\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "VID\_UPD"

(v\_id\_vid VID.id\_vid%TYPE,

v\_vid VID.vid%TYPE)

AS

BEGIN

UPDATE VID set vid= v\_vid

where id\_vid= v\_id\_vid;

END;

--------------------------------------------------------

-- DDL for Procedure ZPD\_INS

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "ZPD\_INS"

(v\_id\_klient ZAEMANE\_POD\_NAEM.id\_klient%TYPE,

v\_id\_slujitel ZAEMANE\_POD\_NAEM.id\_slujitel%TYPE,

v\_data\_na\_zaemane ZAEMANE\_POD\_NAEM.data\_na\_zaemane%TYPE,

v\_broi\_dni ZAEMANE\_POD\_NAEM.broi\_dni%TYPE,

v\_id\_avtomobil ZAEMANE\_POD\_NAEM.ID\_AVTOMOBIL%TYPE)

AS

BEGIN

INSERT INTO ZAEMANE\_POD\_NAEM(ID\_KLIENT, ID\_SLUJITEL, data\_na\_zaemane, broi\_dni, ID\_AVTOMOBIL)

Values (v\_id\_klient, v\_id\_slujitel, v\_data\_na\_zaemane, v\_broi\_dni, v\_id\_avtomobil);

END;

--------------------------------------------------------

-- DDL for Procedure ZPD\_UPD

--------------------------------------------------------

set define off;

CREATE OR REPLACE PROCEDURE "ZPD\_UPD"

(v\_id\_pod\_naem ZAEMANE\_POD\_NAEM.id\_pod\_naem%TYPE,

v\_data\_na\_zaemane ZAEMANE\_POD\_NAEM.data\_na\_zaemane%TYPE,

v\_broi\_dni ZAEMANE\_POD\_NAEM.broi\_dni%TYPE,

v\_id\_klient ZAEMANE\_POD\_NAEM.id\_klient%TYPE,

v\_id\_slujitel ZAEMANE\_POD\_NAEM.id\_slujitel%TYPE,

v\_id\_avtomobil ZAEMANE\_POD\_NAEM.ID\_AVTOMOBIL%TYPE)

AS

BEGIN

UPDATE ZAEMANE\_POD\_NAEM set data\_na\_zaemane= v\_data\_na\_zaemane,

broi\_dni= v\_broi\_dni,

id\_klient = v\_id\_klient,

ID\_SLUJITEL= v\_id\_slujitel,

ID\_AVTOMOBIL= v\_id\_avtomobil

where id\_pod\_naem= v\_id\_pod\_naem;

END;

**5.Sequenc-и за Trigger-ите**

CREATE SEQUENCE "ADR\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence AVT\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "AVT\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence CVQT\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "CVQT\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence KLNT\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "KLNT\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence MDL\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "MDL\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence MRK\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "MRK\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence POS\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "POS\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence SLUJ\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "SLUJ\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence VID\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "VID\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Sequence ZPD\_SEQ

--------------------------------------------------------

CREATE SEQUENCE "ZPD\_SEQ" MINVALUE 1 MAXVALUE 999999999999999999999999999 INCREMENT BY 1 START WITH 21 CACHE 20 NOORDER NOCYCLE

--------------------------------------------------------

-- DDL for Table ADRES

--------------------------------------------------------

CREATE TABLE "ADRES" ("ID\_ADRES" NUMBER(\*,0), "GRAD" CHAR(20))

--------------------------------------------------------

-- DDL for Table AVTOMOBIL

--------------------------------------------------------

CREATE TABLE "AVTOMOBIL" ("ID\_AVTOMOBIL" NUMBER(\*,0), "GODINA" NUMBER(\*,0), "IZMINATI\_KM" NUMBER(\*,0), "CENA\_NA\_DEN" NUMBER(\*,0), "ID\_CVQT" NUMBER(\*,0), "ID\_MODEL" NUMBER(\*,0), "ID\_VID" NUMBER(\*,0))

--------------------------------------------------------

-- DDL for Table CVQT

--------------------------------------------------------

CREATE TABLE "CVQT" ("ID\_CVQT" NUMBER(\*,0), "CVQT" CHAR(50))

--------------------------------------------------------

-- DDL for Table KLIENT

--------------------------------------------------------

CREATE TABLE "KLIENT" ("ID\_KLIENT" NUMBER(\*,0), "IME" CHAR(50), "ULICA" CHAR(50), "TELEFON" NUMBER(\*,0), "ID\_ADRES" NUMBER(\*,0))

--------------------------------------------------------

-- DDL for Table MARKA

--------------------------------------------------------

CREATE TABLE "MARKA" ("ID\_MARKA" NUMBER(\*,0), "MARKA" CHAR(50))

--------------------------------------------------------

-- DDL for Table MODELL

--------------------------------------------------------

CREATE TABLE "MODELL" ("ID\_MODEL" NUMBER(\*,0), "MODEL" CHAR(50), "ID\_MARKA" NUMBER(\*,0))

--------------------------------------------------------

-- DDL for Table MODEL\_TYPE

--------------------------------------------------------

CREATE TABLE "MODEL\_TYPE" ("MODEL\_ID\_MODEL" NUMBER(\*,0), "VID\_ID\_VID" NUMBER(\*,0))

--------------------------------------------------------

-- DDL for Table POZICIQ

--------------------------------------------------------

CREATE TABLE "POZICIQ" ("ID\_POZICIQ" NUMBER(\*,0), "POZICIQ" CHAR(50), "ZAPLATA" VARCHAR2(20))

--------------------------------------------------------

-- DDL for Table SLUJITEL

--------------------------------------------------------

CREATE TABLE "SLUJITEL" ("ID\_SLUJITEL" NUMBER(\*,0), "IME" CHAR(50), "POZICIQ" CHAR(20), "TELEFON" NUMBER(\*,0), "ID\_POZICIQ" NUMBER(\*,0))

--------------------------------------------------------

-- DDL for Table VID

--------------------------------------------------------

CREATE TABLE "VID" ("ID\_VID" NUMBER(\*,0), "VID" CHAR(50))

--------------------------------------------------------

-- DDL for Table ZAEMANE\_POD\_NAEM

--------------------------------------------------------

**6.Trigger-и**

CREATE OR REPLACE TRIGGER "AFTER\_ZAEMANE\_POD\_NAEM\_INS"

AFTER INSERT ON zaemane\_pod\_naem

FOR EACH ROW

BEGIN

UPDATE avtomobil

SET status = '????'

WHERE id\_avtomobil = :NEW.id\_avtomobil;

END;

ALTER TRIGGER "AFTER\_ZAEMANE\_POD\_NAEM\_INS" ENABLE

--------------------------------------------------------

-- DDL for Trigger BEFORE\_ZAEMANE\_POD\_NAEM\_INS

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "BEFORE\_ZAEMANE\_POD\_NAEM\_INS"

BEFORE INSERT ON zaemane\_pod\_naem

FOR EACH ROW

DECLARE

cnt INTEGER;

BEGIN

SELECT COUNT(\*)

INTO cnt

FROM zaemane\_pod\_naem

WHERE id\_klient = :NEW.id\_klient

AND id\_slujitel = :NEW.id\_slujitel

AND data\_na\_zaemane = :NEW.data\_na\_zaemane

AND broi\_dni = :NEW.broi\_dni

AND id\_avtomobil = :NEW.id\_avtomobil;

IF cnt > 0 THEN

RAISE\_APPLICATION\_ERROR(-20103, 'Duplicate rental record');

END IF;

END;

ALTER TRIGGER "BEFORE\_ZAEMANE\_POD\_NAEM\_INS" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_ADRES\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_ADRES\_AUTO\_TRG"

BEFORE INSERT ON ADRES

FOR EACH ROW

WHEN (NEW.id\_adres IS NULL) BEGIN

:NEW.id\_adres := ADR\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_ADRES\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_AVTOMOBIL\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_AVTOMOBIL\_AUTO\_TRG"

BEFORE INSERT ON AVTOMOBIL

FOR EACH ROW

WHEN (NEW.id\_avtomobil IS NULL) BEGIN

:NEW.id\_avtomobil := AVT\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_AVTOMOBIL\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_CVQT\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_CVQT\_AUTO\_TRG"

BEFORE INSERT ON CVQT

FOR EACH ROW

WHEN (NEW.id\_cvqt IS NULL) BEGIN

:NEW.id\_cvqt := CVQT\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_CVQT\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_KLIENT\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_KLIENT\_AUTO\_TRG"

BEFORE INSERT ON KLIENT

FOR EACH ROW

WHEN (NEW.id\_klient IS NULL) BEGIN

:NEW.id\_klient := KLNT\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_KLIENT\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_MARKA\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_MARKA\_AUTO\_TRG"

BEFORE INSERT ON MARKA

FOR EACH ROW

WHEN (NEW.id\_marka IS NULL) BEGIN

:NEW.id\_marka := MRK\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_MARKA\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_MODEL\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_MODEL\_AUTO\_TRG"

BEFORE INSERT ON MODELL

FOR EACH ROW

WHEN (NEW.id\_model IS NULL) BEGIN

:NEW.id\_model := MDL\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_MODEL\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_POD\_NAEM\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_POD\_NAEM\_AUTO\_TRG"

BEFORE INSERT ON ZAEMANE\_POD\_NAEM

FOR EACH ROW

WHEN (NEW.id\_pod\_naem IS NULL) BEGIN

:NEW.id\_pod\_naem := ZPD\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_POD\_NAEM\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_POZICIQ\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_POZICIQ\_AUTO\_TRG"

BEFORE INSERT ON Poziciq

FOR EACH ROW

WHEN (NEW.id\_poziciq IS NULL) BEGIN:NEW.id\_poziciq := POS\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_POZICIQ\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_SLUJITEL\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_SLUJITEL\_AUTO\_TRG"

BEFORE INSERT ON SLUJITEL

FOR EACH ROW

WHEN (NEW.id\_slujitel IS NULL) BEGIN

:NEW.id\_slujitel := SLUJ\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_SLUJITEL\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Trigger ID\_VID\_AUTO\_TRG

--------------------------------------------------------

CREATE OR REPLACE TRIGGER "ID\_VID\_AUTO\_TRG"

BEFORE INSERT ON VID

FOR EACH ROW

WHEN (NEW.id\_vid IS NULL) BEGIN

:NEW.id\_vid := VID\_SEQ.NEXTVAL;

END;

ALTER TRIGGER "ID\_VID\_AUTO\_TRG" ENABLE

--------------------------------------------------------

-- DDL for Procedure ADR\_INS

--------------------------------------------------------

**7.Процедурни справки**

**-отдадени автомобили от служител, подредени по вид и дата**

SELECT

v.vid,

z.data\_na\_zaemane,

a.godina,

a.izminati\_km,

a.cena\_na\_den,

c.cvqt,

m.marka,

mo.model,

s.ime AS slujitel\_ime,

k.ime AS klient\_ime

FROM

zaemane\_pod\_naem z

JOIN

avtomobil a ON z.id\_avtomobil = a.id\_avtomobil

JOIN

vid v ON a.id\_vid = v.id\_vid

JOIN

slujitel s ON z.id\_slujitel = s.id\_slujitel

JOIN

klient k ON z.id\_klient = k.id\_klient

JOIN

cvqt c ON a.id\_cvqt = c.id\_cvqt

JOIN

modell mo ON a.id\_model = mo.id\_model

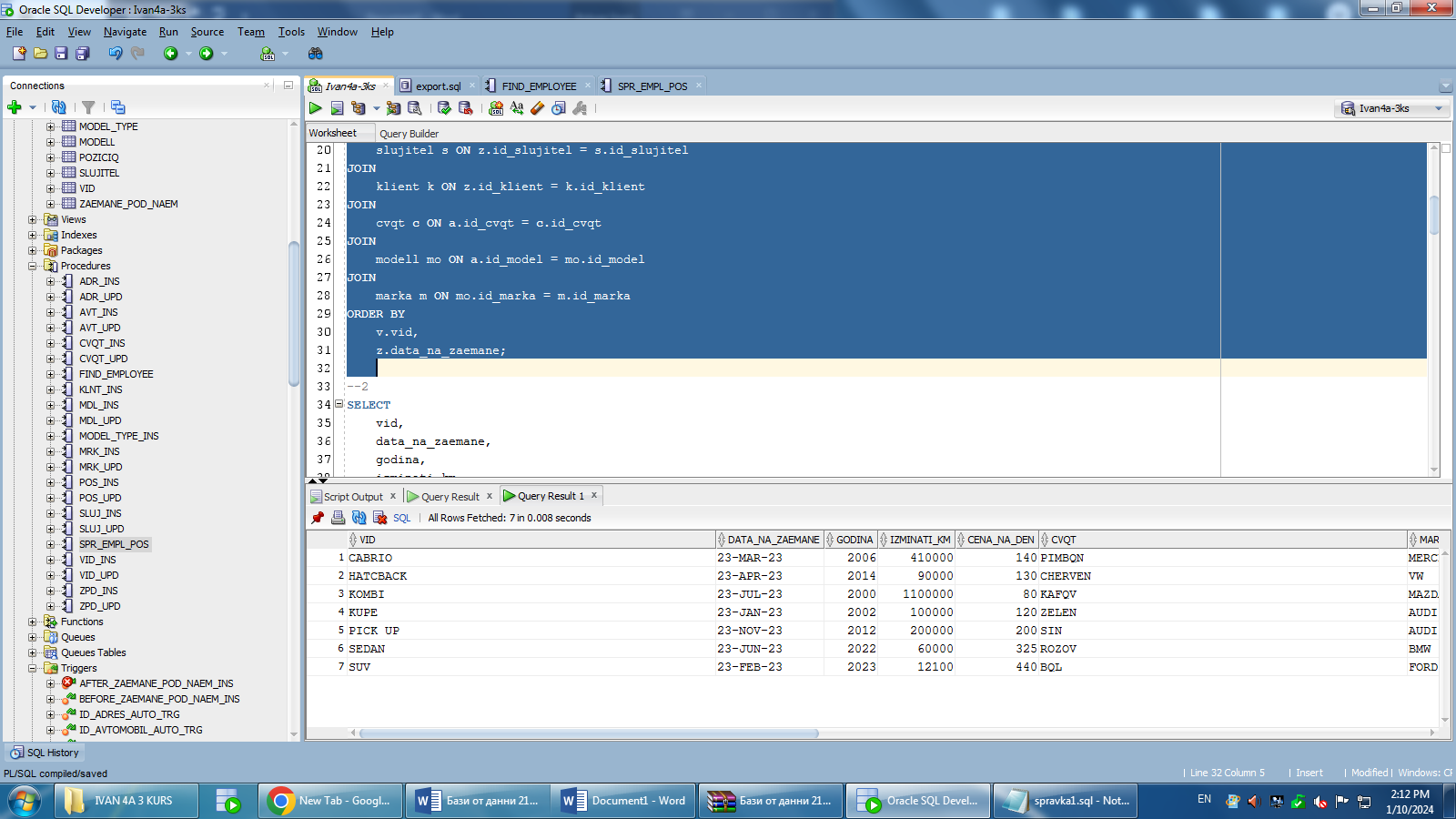
JOIN

marka m ON mo.id\_marka = m.id\_marka

ORDER BY

v.vid,

z.data\_na\_zaemane;



**-последните 10 заемания, подредени по дни**

SELECT

vid,

data\_na\_zaemane,

godina,

izminati\_km,

cena\_na\_den,

cvqt,

marka,

model,

slujitel\_ime,

klient\_ime

FROM (

SELECT

v.vid,

z.data\_na\_zaemane,

a.godina,

a.izminati\_km,

a.cena\_na\_den,

c.cvqt,

m.marka,

mo.model,

s.ime AS slujitel\_ime,

k.ime AS klient\_ime,

ROW\_NUMBER() OVER (ORDER BY z.data\_na\_zaemane DESC) AS rnum

FROM

zaemane\_pod\_naem z

JOIN

avtomobil a ON z.id\_avtomobil = a.id\_avtomobil

JOIN

vid v ON a.id\_vid = v.id\_vid

JOIN

slujitel s ON z.id\_slujitel = s.id\_slujitel

JOIN

klient k ON z.id\_klient = k.id\_klient

JOIN

cvqt c ON a.id\_cvqt = c.id\_cvqt

JOIN

modell mo ON a.id\_model = mo.id\_model

JOIN

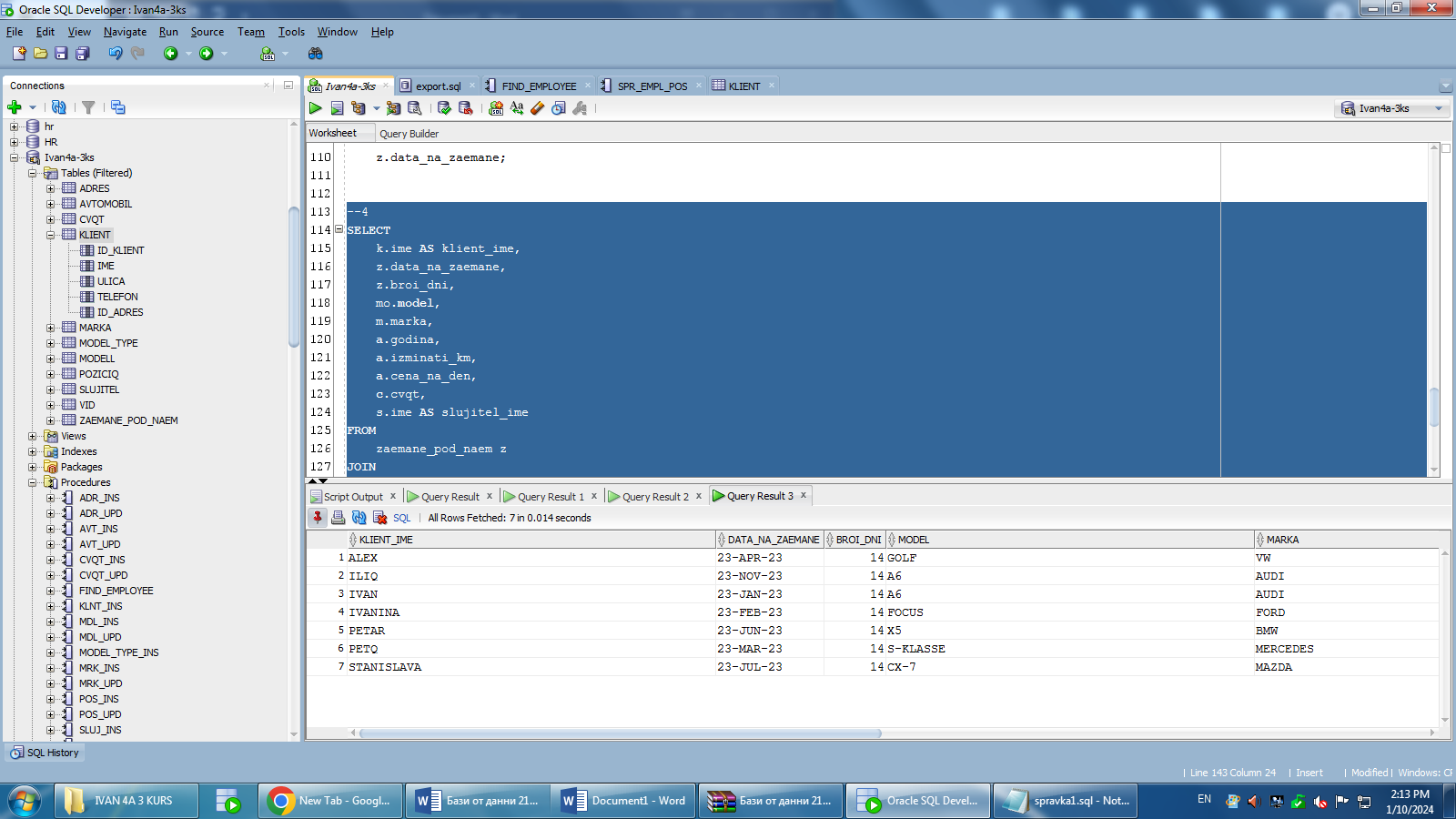
marka m ON mo.id\_marka = m.id\_marka

ORDER BY

z.data\_na\_zaemane DESC

)

WHERE rnum <= 10;



**-наети автомобили от клиент подредени по дата**

SELECT

k.ime AS klient\_ime,

z.data\_na\_zaemane,

z.broi\_dni,

mo.model,

m.marka,

a.godina,

a.izminati\_km,

a.cena\_na\_den,

c.cvqt,

s.ime AS slujitel\_ime

FROM

zaemane\_pod\_naem z

JOIN

avtomobil a ON z.id\_avtomobil = a.id\_avtomobil

JOIN

klient k ON z.id\_klient = k.id\_klient

JOIN

slujitel s ON z.id\_slujitel = s.id\_slujitel

JOIN

cvqt c ON a.id\_cvqt = c.id\_cvqt

JOIN

modell mo ON a.id\_model = mo.id\_model

JOIN

marka m ON mo.id\_marka = m.id\_marka

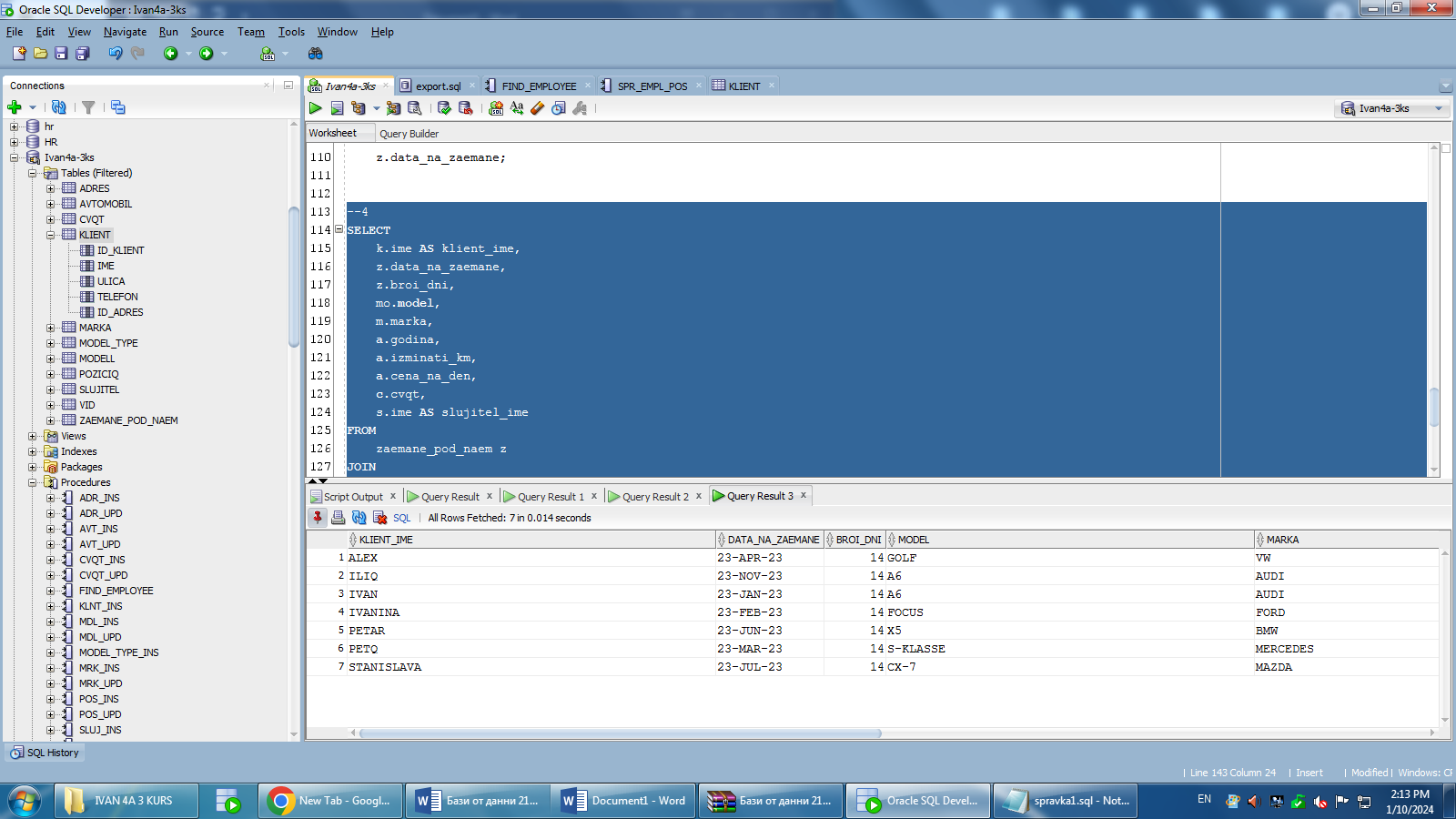
WHERE

z.data\_na\_zaemane BETWEEN TO\_DATE('&начална\_дата', 'DD-MM-RR') AND TO\_DATE('&крайна\_дата', 'DD-MM-RR')

ORDER BY

k.ime,

z.data\_na\_zaemane;



**7.Специален тригер**create or replace TRIGGER before\_zaemane\_pod\_naem\_INS

BEFORE INSERT ON zaemane\_pod\_naem

FOR EACH ROW

DECLARE

cnt INTEGER;

BEGIN

SELECT COUNT(\*)

INTO cnt

FROM zaemane\_pod\_naem

WHERE id\_klient = :NEW.id\_klient

AND id\_slujitel = :NEW.id\_slujitel

AND data\_na\_zaemane = :NEW.data\_na\_zaemane

AND broi\_dni = :NEW.broi\_dni

AND id\_avtomobil = :NEW.id\_avtomobil;

IF cnt > 0 THEN

RAISE\_APPLICATION\_ERROR(-20103, 'Duplicate rental record');

END IF;

END;

create or replace TRIGGER after\_zaemane\_pod\_naem\_INS

AFTER INSERT ON zaemane\_pod\_naem

FOR EACH ROW

BEGIN

UPDATE avtomobil

SET status = '¿¿¿¿'

WHERE id\_avtomobil = :NEW.id\_avtomobil;

END;

**8.Вкарване на данни във таблиците**

REM INSERTING into ADRES

SET DEFINE OFF;

Insert into ADRES (ID\_ADRES,GRAD) values (1,'VARNA ');

Insert into ADRES (ID\_ADRES,GRAD) values (2,'DOBRICH ');

Insert into ADRES (ID\_ADRES,GRAD) values (3,'SILISTRA ');

Insert into ADRES (ID\_ADRES,GRAD) values (4,'KOZLODUI ');

Insert into ADRES (ID\_ADRES,GRAD) values (5,'YAMBOL ');

Insert into ADRES (ID\_ADRES,GRAD) values (6,'POPOVO ');

Insert into ADRES (ID\_ADRES,GRAD) values (7,'BURGAS ');

REM INSERTING into AVTOMOBIL

SET DEFINE OFF;

Insert into AVTOMOBIL (ID\_AVTOMOBIL,GODINA,IZMINATI\_KM,CENA\_NA\_DEN,ID\_CVQT,ID\_MODEL,ID\_VID) values (1,2002,100000,120,1,1,1);

Insert into AVTOMOBIL (ID\_AVTOMOBIL,GODINA,IZMINATI\_KM,CENA\_NA\_DEN,ID\_CVQT,ID\_MODEL,ID\_VID) values (2,2012,200000,200,2,2,2);

Insert into AVTOMOBIL (ID\_AVTOMOBIL,GODINA,IZMINATI\_KM,CENA\_NA\_DEN,ID\_CVQT,ID\_MODEL,ID\_VID) values (3,2022,60000,325,3,3,3);

Insert into AVTOMOBIL (ID\_AVTOMOBIL,GODINA,IZMINATI\_KM,CENA\_NA\_DEN,ID\_CVQT,ID\_MODEL,ID\_VID) values (4,2000,1100000,80,4,4,4);

Insert into AVTOMOBIL (ID\_AVTOMOBIL,GODINA,IZMINATI\_KM,CENA\_NA\_DEN,ID\_CVQT,ID\_MODEL,ID\_VID) values (5,2014,90000,130,5,5,5);

Insert into AVTOMOBIL (ID\_AVTOMOBIL,GODINA,IZMINATI\_KM,CENA\_NA\_DEN,ID\_CVQT,ID\_MODEL,ID\_VID) values (6,2023,12100,440,6,6,6);

Insert into AVTOMOBIL (ID\_AVTOMOBIL,GODINA,IZMINATI\_KM,CENA\_NA\_DEN,ID\_CVQT,ID\_MODEL,ID\_VID) values (7,2006,410000,140,7,7,7);

REM INSERTING into CVQT

SET DEFINE OFF;

Insert into CVQT (ID\_CVQT,CVQT) values (1,'ZELEN ');

Insert into CVQT (ID\_CVQT,CVQT) values (2,'SIN ');

Insert into CVQT (ID\_CVQT,CVQT) values (3,'ROZOV ');

Insert into CVQT (ID\_CVQT,CVQT) values (4,'KAFQV ');

Insert into CVQT (ID\_CVQT,CVQT) values (5,'CHERVEN ');

Insert into CVQT (ID\_CVQT,CVQT) values (6,'BQL ');

Insert into CVQT (ID\_CVQT,CVQT) values (7,'PIMBQN ');

REM INSERTING into KLIENT

SET DEFINE OFF;

Insert into KLIENT (ID\_KLIENT,IME,ULICA,TELEFON,ID\_ADRES) values (1,'IVAN ','STRUMA ',897769464,1);

Insert into KLIENT (ID\_KLIENT,IME,ULICA,TELEFON,ID\_ADRES) values (2,'ILIQ ','KOTELOVA ',897723464,2);

Insert into KLIENT (ID\_KLIENT,IME,ULICA,TELEFON,ID\_ADRES) values (3,'PETAR ','VT ',897545464,3);

Insert into KLIENT (ID\_KLIENT,IME,ULICA,TELEFON,ID\_ADRES) values (4,'STANISLAVA ','STARA PLANINA ',823769464,4);

Insert into KLIENT (ID\_KLIENT,IME,ULICA,TELEFON,ID\_ADRES) values (5,'ALEX ','BELOSLAV ',897725564,5);

Insert into KLIENT (ID\_KLIENT,IME,ULICA,TELEFON,ID\_ADRES) values (6,'IVANINA ','KOZLODUI ',892269464,6);

Insert into KLIENT (ID\_KLIENT,IME,ULICA,TELEFON,ID\_ADRES) values (7,'PETQ ','CARIGRADSKO ',897769411,7);

REM INSERTING into MARKA

SET DEFINE OFF;

Insert into MARKA (ID\_MARKA,MARKA) values (1,'AUDI ');

Insert into MARKA (ID\_MARKA,MARKA) values (2,'BMW ');

Insert into MARKA (ID\_MARKA,MARKA) values (3,'MAZDA ');

Insert into MARKA (ID\_MARKA,MARKA) values (4,'VW ');

Insert into MARKA (ID\_MARKA,MARKA) values (5,'FORD ');

Insert into MARKA (ID\_MARKA,MARKA) values (6,'MERCEDES ');

Insert into MARKA (ID\_MARKA,MARKA) values (7,'ISUZU ');

REM INSERTING into MODELL

SET DEFINE OFF;

Insert into MODELL (ID\_MODEL,MODEL,ID\_MARKA) values (1,'A6 ',1);

Insert into MODELL (ID\_MODEL,MODEL,ID\_MARKA) values (2,'A6 ',1);

Insert into MODELL (ID\_MODEL,MODEL,ID\_MARKA) values (3,'X5 ',2);

Insert into MODELL (ID\_MODEL,MODEL,ID\_MARKA) values (4,'CX-7 ',3);

Insert into MODELL (ID\_MODEL,MODEL,ID\_MARKA) values (5,'GOLF ',4);

Insert into MODELL (ID\_MODEL,MODEL,ID\_MARKA) values (6,'FOCUS ',5);

Insert into MODELL (ID\_MODEL,MODEL,ID\_MARKA) values (7,'S-KLASSE ',6);

REM INSERTING into MODEL\_TYPE

SET DEFINE OFF;

REM INSERTING into POZICIQ

SET DEFINE OFF;

Insert into POZICIQ (ID\_POZICIQ,POZICIQ,ZAPLATA) values (1,'BOSS ','10000');

Insert into POZICIQ (ID\_POZICIQ,POZICIQ,ZAPLATA) values (2,'CHISTACH ','800');

Insert into POZICIQ (ID\_POZICIQ,POZICIQ,ZAPLATA) values (3,'CHETOVODITEL ','2000');

Insert into POZICIQ (ID\_POZICIQ,POZICIQ,ZAPLATA) values (4,'UPRAVITEL ','4000');

Insert into POZICIQ (ID\_POZICIQ,POZICIQ,ZAPLATA) values (5,'TEHNIK ','1200');

Insert into POZICIQ (ID\_POZICIQ,POZICIQ,ZAPLATA) values (6,'RABOTA S KLIENTI ','2100');

Insert into POZICIQ (ID\_POZICIQ,POZICIQ,ZAPLATA) values (7,'FRONT OFFICE ','2200');

REM INSERTING into SLUJITEL

SET DEFINE OFF;

Insert into SLUJITEL (ID\_SLUJITEL,IME,POZICIQ,TELEFON,ID\_POZICIQ) values (1,'ILIAN ','BOSS ',897714264,1);

Insert into SLUJITEL (ID\_SLUJITEL,IME,POZICIQ,TELEFON,ID\_POZICIQ) values (2,'SUBI ','CHISTACH ',897124214,2);

Insert into SLUJITEL (ID\_SLUJITEL,IME,POZICIQ,TELEFON,ID\_POZICIQ) values (3,'PETAR ','CHETOVODITEL ',897741264,3);

Insert into SLUJITEL (ID\_SLUJITEL,IME,POZICIQ,TELEFON,ID\_POZICIQ) values (4,'IVANINA ','UPRAVITEL ',824114264,4);

Insert into SLUJITEL (ID\_SLUJITEL,IME,POZICIQ,TELEFON,ID\_POZICIQ) values (5,'MARINA ','TEHNIK ',892314264,5);

Insert into SLUJITEL (ID\_SLUJITEL,IME,POZICIQ,TELEFON,ID\_POZICIQ) values (6,'ISMA ','RABOTA ',872414264,6);

Insert into SLUJITEL (ID\_SLUJITEL,IME,POZICIQ,TELEFON,ID\_POZICIQ) values (7,'ILARION ','FRONT ',897244264,7);

REM INSERTING into VID

SET DEFINE OFF;

Insert into VID (ID\_VID,VID) values (1,'KUPE ');

Insert into VID (ID\_VID,VID) values (2,'PICK UP ');

Insert into VID (ID\_VID,VID) values (3,'SEDAN ');

Insert into VID (ID\_VID,VID) values (4,'KOMBI ');

Insert into VID (ID\_VID,VID) values (5,'HATCBACK ');

Insert into VID (ID\_VID,VID) values (6,'SUV ');

Insert into VID (ID\_VID,VID) values (7,'CABRIO ');

REM INSERTING into ZAEMANE\_POD\_NAEM

SET DEFINE OFF;

Insert into ZAEMANE\_POD\_NAEM (ID\_POD\_NAEM,ID\_KLIENT,ID\_SLUJITEL,DATA\_NA\_ZAEMANE,BROI\_DNI,ID\_AVTOMOBIL) values (2,1,1,to\_date('23-JAN-23','DD-MON-RR'),14,1);

Insert into ZAEMANE\_POD\_NAEM (ID\_POD\_NAEM,ID\_KLIENT,ID\_SLUJITEL,DATA\_NA\_ZAEMANE,BROI\_DNI,ID\_AVTOMOBIL) values (3,2,2,to\_date('23-NOV-23','DD-MON-RR'),14,2);

Insert into ZAEMANE\_POD\_NAEM (ID\_POD\_NAEM,ID\_KLIENT,ID\_SLUJITEL,DATA\_NA\_ZAEMANE,BROI\_DNI,ID\_AVTOMOBIL) values (4,3,3,to\_date('23-JUN-23','DD-MON-RR'),14,3);

Insert into ZAEMANE\_POD\_NAEM (ID\_POD\_NAEM,ID\_KLIENT,ID\_SLUJITEL,DATA\_NA\_ZAEMANE,BROI\_DNI,ID\_AVTOMOBIL) values (5,4,4,to\_date('23-JUL-23','DD-MON-RR'),14,4);

Insert into ZAEMANE\_POD\_NAEM (ID\_POD\_NAEM,ID\_KLIENT,ID\_SLUJITEL,DATA\_NA\_ZAEMANE,BROI\_DNI,ID\_AVTOMOBIL) values (6,5,5,to\_date('23-APR-23','DD-MON-RR'),14,5);

Insert into ZAEMANE\_POD\_NAEM (ID\_POD\_NAEM,ID\_KLIENT,ID\_SLUJITEL,DATA\_NA\_ZAEMANE,BROI\_DNI,ID\_AVTOMOBIL) values (7,6,6,to\_date('23-FEB-23','DD-MON-RR'),14,6);

Insert into ZAEMANE\_POD\_NAEM (ID\_POD\_NAEM,ID\_KLIENT,ID\_SLUJITEL,DATA\_NA\_ZAEMANE,BROI\_DNI,ID\_AVTOMOBIL) values (8,7,7,to\_date('23-MAR-23','DD-MON-RR'),14,7);

--------------------------------------------------------

-- DDL for Index RELATION\_16\_PK

--------------------------------------------------------