Технически университет – Варна

**Дисциплина:** *Бази от Данни*

ДОКУМЕНТАЦИЯ НА

ЗАДАНИЕ

**Ръководител:**

*гл. ас. др. инж. Диян Динев*

**Информация за Студента:**

**Име:** *Даниел Димитров*

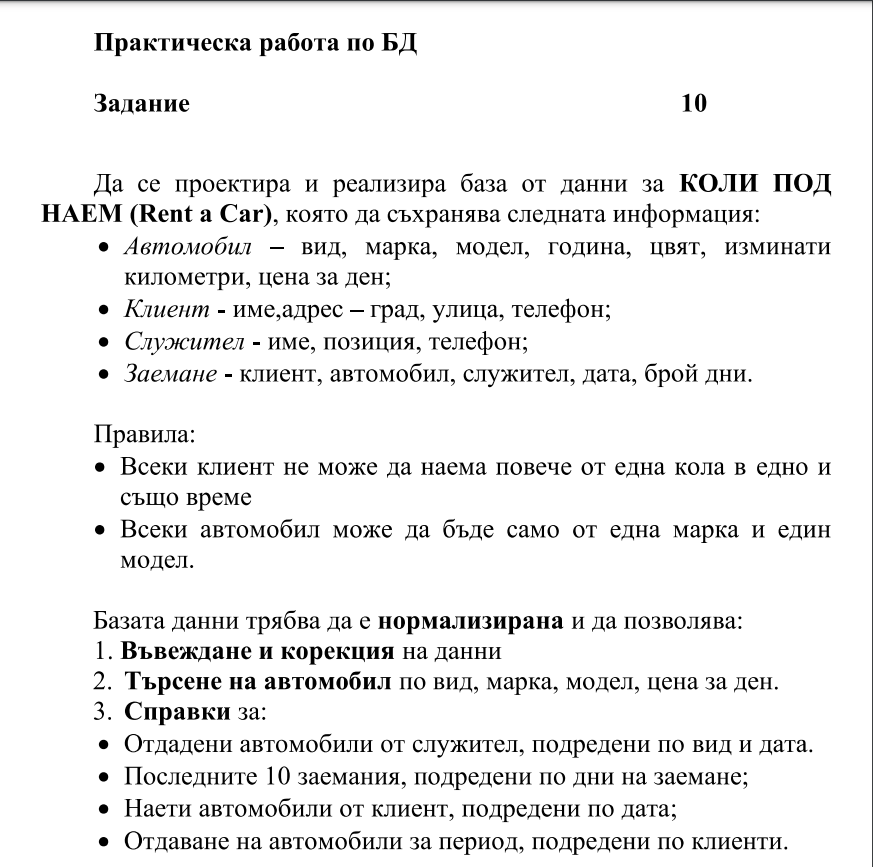
**Фак. Номер:** *19623345*

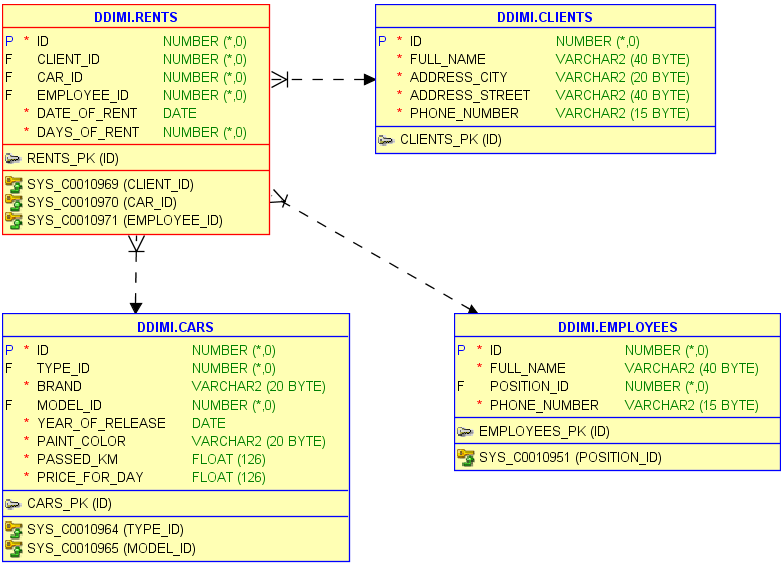
**ГРУПА: 1**

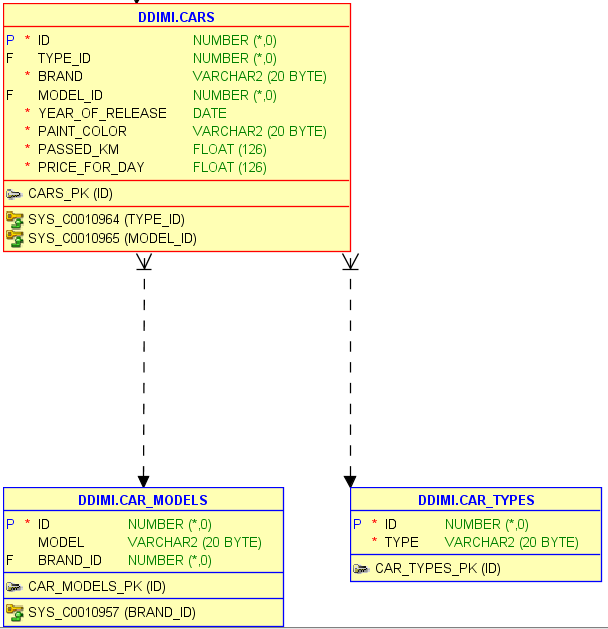
**Факултет:** *ФИТА*

**Катедра:** *КНТ*

**Специалност:** *КСТ*

1. **Задание:**
2. **Модели:**





1. **Създаване на таблици:**

create table clients(

    id int,

    full\_name varchar(40) not null,

    address\_city varchar(20) not null,

    address\_street varchar(40) not null,

    phone\_number varchar(15) not null,

    primary key(id)

);

create table employee\_positions(

    id int,

    position varchar(20) not null,

    primary key(id)

);

create table employees(

    id int,

    full\_name varchar(40) not null,

    position\_id references employee\_positions(id),

    phone\_number varchar(15) not null,

    primary key(id)

);

create table car\_types(

    id int,

    type varchar(20) not null,

    primary key(id)

);

create table car\_brands(

    id int,

    brand varchar(20) not null,

    primary key(id)

);

create table car\_models(

    id int,

    model varchar(20),

    brand\_id references car\_brands(id),

    primary key(id)

);

create table cars(

    id int,

**type\_id** references car\_types(id),

    brand varchar(20) not null,

    model\_id references car\_models(id),

    year\_of\_release date not null,

    paint\_color varchar(20) not null,

    passed\_km float not null,

    price\_for\_day float not null,

    primary key(id)

);

create table rents(

    id int,

    client\_id references clients(id),

    car\_id references cars(id),

    employee\_id references employees(id),

    date\_of\_rent date not null,

    days\_of\_rent int not null,

    primary key(id)

);

1. **Тригери:**

begin

    for i in (select sequence\_name from USER\_SEQUENCES)

    loop

        execute immediate ('drop sequence '|| i.sequence\_name);

    end loop;

end;

/

create sequence clients\_seq start with 1 increment by 1 nocache order;

create sequence employee\_positions\_seq start with 1 increment by 1 nocache order;

create sequence employees\_seq start with 1 increment by 1 nocache order;

create sequence car\_types\_seq start with 1 increment by 1 nocache order;

create sequence car\_brands\_seq start with 1 increment by 1 nocache order;

create sequence car\_models\_seq  start with 1 increment by 1 nocache order;

create sequence cars\_seq start with 1 increment by 1 nocache order;

create sequence rents\_seq start with 1 increment by 1 nocache order;

*/\**

*Known error:*

*https://logbuffer.wordpress.com/2011/10/19/oracle-ora-0600-on-creating-trigger-on-11-2/*

*\*/*

ALTER SESSION SET PLSCOPE\_SETTINGS = 'IDENTIFIERS:NONE';

create or **replace** trigger clients\_trg

before insert on clients

for each row when(new.id is null)

begin

    :new.id := clients\_seq.nextval;

end;

/

create or **replace** trigger employee\_positions\_trg

before insert on employee\_positions

for each row when(new.id is null)

begin

    :new.id := employee\_positions\_seq.nextval;

end;

/

create or **replace** trigger employees\_trg

before insert on employees

for each row when(new.id is null)

begin

    :new.id := employees\_seq.nextval;

end;

/

create or **replace** trigger car\_types\_trg

before insert on car\_types

for each row when(new.id is null)

begin

    :new.id := car\_types\_seq.nextval;

end;

/

create or **replace** trigger car\_brands\_trg

before insert on car\_brands

for each row when(new.id is null)

begin

    :new.id := car\_brands\_seq.nextval;

end;

/

create or **replace** trigger car\_models\_trg

before insert on car\_models

for each row when(new.id is null)

begin

    :new.id := car\_models\_seq.nextval;

end;

/

create or **replace** trigger cars\_trg

before insert on cars

for each row when(new.id is null)

begin

    :new.id := cars\_seq.nextval;

end;

/

create or **replace** trigger rents\_trg

before insert on rents

for each row when(new.id is null)

begin

    :new.id := rents\_seq.nextval;

end;

/ALTER SESSION SET PLSCOPE\_SETTINGS = 'IDENTIFIERS:ALL';

1. **Процедури:**

CREATE OR **REPLACE** PROCEDURE clients\_insert (

    full\_name        clients.full\_name%TYPE,

    address\_city     clients.address\_city%TYPE,

    address\_street   clients.address\_street%TYPE,

    phone\_number     clients.phone\_number%TYPE

) AS

BEGIN

    INSERT INTO clients VALUES (

        NULL,

        full\_name,

        address\_city,

        address\_street,

        phone\_number

    );

END;

/

CREATE OR **REPLACE** PROCEDURE employee\_positions\_insert (

    position employee\_positions.position%TYPE

) AS

BEGIN

    INSERT INTO employee\_positions VALUES (

        NULL,

        position

    );

END;

/

CREATE OR **REPLACE** PROCEDURE employees\_insert (

    full\_name      employees.full\_name%TYPE,

    position\_id    employees.position\_id%TYPE,

    phone\_number   employees.phone\_number%TYPE

) AS

BEGIN

    INSERT INTO employees VALUES (

        NULL,

        full\_name,

        position\_id,

        phone\_number

    );

END;

/

CREATE OR **REPLACE** PROCEDURE car\_types\_insert (

    type car\_types.type%TYPE

) AS

BEGIN

    INSERT INTO car\_types VALUES (

        NULL,

        type

    );

END;

/

CREATE OR **REPLACE** PROCEDURE car\_brands\_insert (

    brand car\_brands.brand%TYPE

) AS

BEGIN

    INSERT INTO car\_brands VALUES (

        NULL,

        brand

    );

END;

/

CREATE OR **REPLACE** PROCEDURE car\_models\_insert (

    model      car\_models.model%TYPE,

    brand\_id   car\_models.brand\_id%TYPE

) AS

BEGIN

    INSERT INTO car\_models VALUES (

        NULL,

        model,

        brand\_id

    );

END;

/

CREATE OR **REPLACE** PROCEDURE cars\_insert (

**type\_id**           cars.type\_id%TYPE,

    brand             cars.brand%TYPE,

    model\_id          cars.model\_id%TYPE,

    year\_of\_release   cars.year\_of\_release%TYPE,

    paint\_color       cars.paint\_color%TYPE,

    passed\_km         cars.passed\_km%TYPE,

    price\_for\_day     cars.price\_for\_day%TYPE

) AS

BEGIN

    INSERT INTO cars VALUES (

        NULL,

**type\_id**,

        brand,

        model\_id,

        year\_of\_release,

        paint\_color,

        passed\_km,

        price\_for\_day

    );

END;

/

CREATE OR **REPLACE** PROCEDURE rents\_insert (

    client\_id      rents.client\_id%TYPE,

    car\_id         rents.car\_id%TYPE,

    employee\_id    rents.employee\_id%TYPE,

    date\_of\_rent   rents.date\_of\_rent%TYPE,

    days\_of\_rent   rents.days\_of\_rent%TYPE

) AS

BEGIN

    INSERT INTO rents VALUES (

        NULL,

        client\_id,

        car\_id,

        employee\_id,

        date\_of\_rent,

        days\_of\_rent

    );

END;

/

CREATE OR **REPLACE** PROCEDURE clients\_delete(id\_to\_delete clients.id%type) AS

BEGIN

    DELETE FROM clients WHERE id = id\_to\_delete;

END;

/

CREATE OR **REPLACE** PROCEDURE employee\_positions\_delete(id\_to\_delete employee\_positions.id%type) AS

BEGIN

    DELETE FROM employee\_positions WHERE id = id\_to\_delete;

END;

/

CREATE OR **REPLACE** PROCEDURE employees\_delete(id\_to\_delete employees.id%type) as

begin

    DELETE FROM employees WHERE id = id\_to\_delete;

END;

/

CREATE OR **REPLACE** PROCEDURE car\_types\_delete(id\_to\_delete car\_types.id%type) AS

BEGIN

    DELETE FROM car\_types WHERE id = id\_to\_delete;

END;

/

CREATE OR **REPLACE** PROCEDURE car\_brands\_delete(id\_to\_delete car\_brands.id%type) AS

BEGIN

    DELETE FROM car\_brands WHERE id = id\_to\_delete;

END;

/

CREATE OR **REPLACE** PROCEDURE car\_models\_delete(id\_to\_delete car\_models.id%type) AS

BEGIN

    DELETE FROM car\_models WHERE id = id\_to\_delete;

END;

/

CREATE OR **REPLACE** PROCEDURE cars\_delete(id\_to\_delete cars.id%type) AS

BEGIN

    DELETE FROM cars WHERE id = id\_to\_delete;

END;

/

CREATE OR **REPLACE** PROCEDURE rents\_delete(id\_to\_delete rents.id%type) AS

BEGIN

    DELETE FROM rents WHERE id = id\_to\_delete;

END;

/

CREATE OR **REPLACE** PROCEDURE clients\_update (

    id\_to\_update         clients.id%TYPE,

    new\_full\_name        clients.full\_name%TYPE,

    new\_address\_city     clients.address\_city%TYPE,

    new\_address\_street   clients.address\_street%TYPE,

    new\_phone\_number     clients.phone\_number%TYPE

) AS

BEGIN

    UPDATE clients

    SET

        full\_name = new\_full\_name,

        address\_city = new\_address\_city,

        address\_street = new\_address\_street,

        phone\_number = new\_phone\_number

    WHERE

        id = id\_to\_update;

END;

/

CREATE OR **REPLACE** PROCEDURE employee\_positions\_update ( id\_to\_update employee\_positions.id%type,

new\_position employee\_positions.positions%TYPE

) AS

BEGIN

    UPDATE employee\_positions

    SET

        position = new\_position

    WHERE

        id = id\_to\_update;

END;

/

CREATE OR **REPLACE** PROCEDURE employees\_update (

    id\_to\_update       employees.id%TYPE,

    new\_full\_name      employees.full\_name%TYPE,

    new\_position\_id    employees.position\_id%TYPE,

    new\_phone\_number   employees.phone\_number%TYPE

) AS

BEGIN

    UPDATE employees

    SET

        full\_name = new\_full\_name,

        position\_id = new\_position\_id,

        phone\_number = new\_phone\_number

    WHERE

        id = id\_to\_update;

END;

/

CREATE OR **REPLACE** PROCEDURE car\_types\_update (

    id\_to\_update car\_types.id%type,

    new\_type car\_types.type%type

) AS

BEGIN

    UPDATE car\_types

    SET

        type = new\_type

    WHERE

        id = id\_to\_update;

END;

/

CREATE OR **REPLACE** PROCEDURE car\_brands\_update (

    id\_to\_update car\_brands.id%type,

    new\_brand car\_brands.brand%type

) AS

BEGIN

    UPDATE car\_brands

    SET

        brand = new\_brand

    WHERE

        id = id\_to\_update;

END;

/

CREATE OR **REPLACE** PROCEDURE car\_models\_update (

    id\_to\_update car\_models.id%type,

    new\_model car\_models.model%type,

    new\_brand\_id car\_models.brand\_id%type

) AS

BEGIN

    UPDATE car\_models

    SET

        model = new\_model,

        brand\_id = new\_brand\_id

    WHERE

        id = id\_to\_update;

END;

/

CREATE OR **REPLACE** PROCEDURE cars\_update (

    id\_to\_update cars.id%type,

    new\_type\_id cars.type\_id%type,

    new\_brand cars.brand%type,

    new\_model\_id cars.model\_id%type,

    new\_year\_of\_release cars.year\_of\_release%type,

    new\_paint\_color cars.paint\_color%type,

    new\_passed\_km cars.passed\_km%type,

    new\_price\_for\_day cars.price\_for\_day%type

) AS

BEGIN

    UPDATE cars

    SET

**type\_id** = new\_type\_id,

        brand = new\_brand,

        model\_id = new\_model\_id,

        year\_of\_release = new\_year\_of\_release,

        paint\_color = new\_paint\_color,

        passed\_km = new\_passed\_km,

        price\_for\_day = new\_price\_for\_day

    WHERE

        id = id\_to\_update;

END;

/

CREATE OR **REPLACE** PROCEDURE rents\_update (

    id\_to\_update rents.id%type,

    new\_client\_id rents.client\_id%type,

    new\_car\_id rents.car\_id%type,

    new\_employee\_id rents.employee\_id%type,

    new\_date\_of\_rent rents.date\_of\_rent%type,

    new\_days\_of\_rent rents.days\_of\_rent%type

) AS

BEGIN

    UPDATE rents

    SET

    client\_id = new\_client\_id,

    car\_id = new\_car\_id,

    employee\_id = new\_employee\_id,

    date\_of\_rent = new\_date\_of\_rent,

    days\_of\_rent = new\_days\_of\_rent

WHERE

    id = id\_to\_update;

END;

/

1. **Примерни данни:**

begin

    employee\_positions\_insert('director');

    employee\_positions\_insert('products manager');

    employee\_positions\_insert('executive');

    employee\_positions\_insert('seller');

    employees\_insert('Georgi Dimitrov', 1, '0880099320');

    employees\_insert('Stefan Georgiev', 2, '0881199321');

    employees\_insert('Petur Ivanov', 3, '0882299322');

    employees\_insert('Petur Petrov', 4, '0883399323');

    employees\_insert('Martin Stamboliiski', 4, '0884499324');

    employees\_insert('Angel Stefanov', 4, '0885599325');

    employees\_insert('Yoan Qnev', 4, '0886699326');

    clients\_insert('Daniel Dimitrov', 'Burgas', 'Izgrev', '0887575232');

    clients\_insert('Ivailo Vasilev', 'Ruse', 'Bqla', '0892134455');

    clients\_insert('Dimitur Zurlev', 'Turgovishte', 'Rusenska 4', '08770774666');

    clients\_insert('Nikolai Zhelev', 'Shumen', 'Stefan Stambolov 12', '08770774563');

    clients\_insert('Dimitur Stoqnov', 'Varna', 'Voivoda 59', '08770774669');

    clients\_insert('Aleksandur Vasilev', 'Plovdiv', 'Centur 99', '08770774667');

    clients\_insert('Georgi Evtimov', 'Sofiq', 'Manastirski Livadi', '08770774663');

    car\_types\_insert('Sedan');

    car\_types\_insert('Kombi');

    car\_types\_insert('Hatchback');

    car\_types\_insert('SUV');

    car\_types\_insert('MPV');

    car\_types\_insert('CUV');

    car\_types\_insert('VAN');

    car\_types\_insert('TRUCK/PICKUP');

    car\_types\_insert('Limousine');

    car\_brands\_insert('Toyota');

    car\_brands\_insert('Ford');

    car\_brands\_insert('Honda');

    car\_brands\_insert('Alfa Romeo');

    car\_brands\_insert('Volvo');

    car\_brands\_insert('VW/Volkswagen');

    car\_brands\_insert('Peugeot');

    car\_brands\_insert('Audi');

    car\_brands\_insert('BMW');

    car\_brands\_insert('Citroen');

    car\_models\_insert('Auris', 1);

    car\_models\_insert('Yaris', 1);

    car\_models\_insert('Corolla', 1);

    car\_models\_insert('Focus', 2);

    car\_models\_insert('Fiesta', 2);

    car\_models\_insert('Taurus', 2);

    car\_models\_insert('Accord', 3);

    car\_models\_insert('Civic', 3);

    car\_models\_insert('CR-V', 3);

    car\_models\_insert('Giulietta 5 gen', 4);

    car\_models\_insert('4C Spider 1 gen', 4);

    car\_models\_insert('AR-146', 4);

    car\_models\_insert('XC60', 5);

    car\_models\_insert('S60', 5);

    car\_models\_insert('V40', 5);

    car\_models\_insert('Golf 3', 6);

    car\_models\_insert('Golf 4', 6);

    car\_models\_insert('Golf 5', 6);

    car\_models\_insert('207', 7);

    car\_models\_insert('307', 7);

    car\_models\_insert('407', 7);

    car\_models\_insert('A3', 8);

    car\_models\_insert('A6', 8);

    car\_models\_insert('A8', 8);

    car\_models\_insert('E30 M3', 9);

    car\_models\_insert('i8', 9);

    car\_models\_insert('X1', 9);

    car\_models\_insert('C5', 10);

    car\_models\_insert('C4', 10);

    car\_models\_insert('C3', 10);

    cars\_insert(3, 'Toyota', 1, to\_date('06/05/2008', 'dd/mm/yyyy'), 'metalic', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Toyota', 2, to\_date('21/01/2009', 'dd/mm/yyyy'), 'black', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Toyota', 3, to\_date('26/04/2003', 'dd/mm/yyyy'), 'white', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(2, 'Ford', 4, to\_date('01/06/2005', 'dd/mm/yyyy'), 'silver', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Ford', 5, to\_date('02/02/2001', 'dd/mm/yyyy'), 'white', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(4, 'Ford', 6, to\_date('26/04/2003', 'dd/mm/yyyy'), 'royal blue', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(1, 'Honda', 7, to\_date('15/09/2004', 'dd/mm/yyyy'), 'gray mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Honda', 8, to\_date('06/09/2011', 'dd/mm/yyyy'), 'dark mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(4, 'Honda', 9, to\_date('17/07/2008', 'dd/mm/yyyy'), 'red', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(1, 'Alfa Romeo', 10, to\_date('04/10/2011', 'dd/mm/yyyy'), 'light gray mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Alfa Romeo', 11, to\_date('29/11/2014', 'dd/mm/yyyy'), 'red mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Alfa Romeo', 12, to\_date('11/08/2008', 'dd/mm/yyyy'), 'black metalic', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(4, 'Volvo', 13, to\_date('09/10/2020', 'dd/mm/yyyy'), 'royal blue mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(1, 'Volvo', 14, to\_date('07/07/2016', 'dd/mm/yyyy'), 'white', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Volvo', 15, to\_date('09/09/2015', 'dd/mm/yyyy'), 'light blue metalic', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'VW', 16, to\_date('23/10/1994', 'dd/mm/yyyy'), 'black', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'VW', 17, to\_date('01/09/2005', 'dd/mm/yyyy'), 'gray', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'VW', 18, to\_date('09/09/2004', 'dd/mm/yyyy'), 'dark blue mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Peugeot', 19, to\_date('23/05/2008', 'dd/mm/yyyy'), 'black', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Peugeot', 20, to\_date('30/09/2005', 'dd/mm/yyyy'), 'gray mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Peugeot', 21, to\_date('17/01/2006', 'dd/mm/yyyy'), 'blue', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Audi', 22, to\_date('23/02/2007', 'dd/mm/yyyy'), 'black', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(1, 'Audi', 23, to\_date('30/05/2004', 'dd/mm/yyyy'), 'gray mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(1, 'Audi', 24, to\_date('17/01/2010', 'dd/mm/yyyy'), 'dark blue', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'BMW', 25, to\_date('23/02/2012', 'dd/mm/yyyy'), 'black', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(1, 'BMW', 26, to\_date('30/05/2013', 'dd/mm/yyyy'), 'gray mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(1, 'BMW', 27, to\_date('17/01/2017', 'dd/mm/yyyy'), 'dark blue', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(3, 'Citroen', 28, to\_date('23/02/2016', 'dd/mm/yyyy'), 'black', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(4, 'Citroen', 29, to\_date('30/05/2003', 'dd/mm/yyyy'), 'gray mate', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    cars\_insert(1, 'Citroen', 30, to\_date('17/01/2012', 'dd/mm/yyyy'), 'dark blue', **round**(dbms\_random.value(50000, 300000), 2), **round**(dbms\_random.value(10, 99), 2));

    rents\_insert(1, 30, 4, to\_date('23/02/2022', 'dd/mm/yyyy'), **floor**(dbms\_random.value(7, 31)));

    rents\_insert(2, 8, 2, to\_date('30/01/2022', 'dd/mm/yyyy'), 31);

    rents\_insert(3, 11, 3, to\_date('15/02/2022', 'dd/mm/yyyy'), 15);

    rents\_insert(4, 16, 5, to\_date('05/03/2022', 'dd/mm/yyyy'), 21);

    rents\_insert(5, 4, 5, to\_date('01/04/2022', 'dd/mm/yyyy'), 6);

    rents\_insert(6, 1, 6, to\_date('28/02/2022', 'dd/mm/yyyy'), 7);

    rents\_insert(7, 21, 1, to\_date('27/02/2022', 'dd/mm/yyyy'), 3);

end;

1. **Справка „Търсене на автомобил по вид, марка, модел, цена за ден“**

* **Процедура**

create or **replace** PROCEDURE search\_car (

    s\_car\_brand      cars.brand%TYPE,

    s\_car\_model       car\_models.model%TYPE,

    s\_car\_type        car\_types.type%TYPE,

    s\_price\_for\_day   cars.price\_for\_day%TYPE

)

IS

BEGIN

    DECLARE

        CURSOR car\_rows IS

        SELECT

            brand,

            car\_models.model,

            car\_types.type,

            year\_of\_release,

            paint\_color,

            passed\_km,

            price\_for\_day

        FROM

            cars

            INNER JOIN car\_models ON **upper**(car\_models.model) = **upper**(s\_car\_model)

            INNER JOIN car\_types ON **upper**(car\_types.type) = **upper**(s\_car\_type)

        WHERE

**upper**(brand) = **upper**(s\_car\_brand)

            AND price\_for\_day >= s\_price\_for\_day - 15

            AND price\_for\_day <= s\_price\_for\_day + 15

        ORDER BY price\_for\_day ASC;

    BEGIN

        dbms\_output.put\_line('Search Results for: ' || s\_car\_brand || ' ' || s\_car\_model);

        FOR car IN car\_rows LOOP

            dbms\_output.put\_line('Brand: ' || car.brand);

            dbms\_output.put\_line('Model: ' || car.model);

            dbms\_output.put\_line('YearOfRelease: ' || car.year\_of\_release);

            dbms\_output.put\_line('Paint: ' || car.paint\_color);

            dbms\_output.put\_line('PriceForDay: ' || car.price\_for\_day);

            dbms\_output.put\_line('PassedKm: ' || car.passed\_km);

            dbms\_output.put\_line('-----------');

        END LOOP;

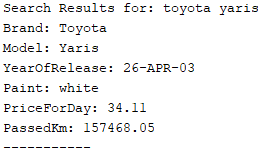
    END;

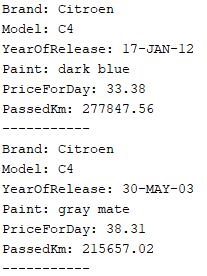
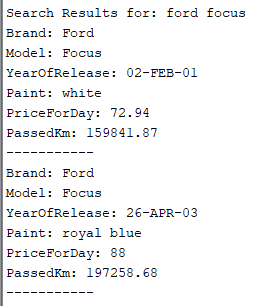
END;

* **Резултати**

set serveroutput on;

exec search\_car('&CAR\_BRAND', '&CAR\_MODEL', '&CAR\_TYPE', &PRICE\_FOR\_DAY);





1. **Справка „Отдадени автомобили от служител, подредени по вид и дата“:**

* **Процедура**

create or **replace** PROCEDURE search\_car\_rentals\_by\_employee (

    s\_employee\_full\_name employees.full\_name%TYPE

) IS

BEGIN

    DECLARE

        CURSOR car\_rental\_rows IS

        SELECT

            clients.full\_name     AS client\_full\_name,

            cars.brand as car\_brand,

            car\_models.model as car\_model,

            car\_types.type as car\_type,

            date\_of\_rent,

            days\_of\_rent

        FROM

            rents

            INNER JOIN cars ON cars.id = car\_id

            INNER JOIN car\_models ON cars.model\_id = car\_models.id

            INNER JOIN car\_types ON cars.type\_id = car\_types.id

            INNER JOIN clients ON clients.id = client\_id

            INNER JOIN employees ON employees.id = employee\_id

        WHERE

**upper**(employees.full\_name) = **upper**(s\_employee\_full\_name)

        ORDER BY date\_of\_rent DESC, car\_types.type ASC;

    BEGIN

        dbms\_output.put\_line('Car rentals from Employee: ' || s\_employee\_full\_name);

        FOR car\_rental IN car\_rental\_rows LOOP

            dbms\_output.put\_line('Client: ' || car\_rental.client\_full\_name);

            dbms\_output.put\_line('Car: ' || car\_rental.car\_brand || ' ' || car\_rental.car\_model || ' ' || car\_rental.car\_type);

            dbms\_output.put\_line('DateOfRent: ' || car\_rental.date\_of\_rent);

            dbms\_output.put\_line('DaysOfRent: ' || car\_rental.days\_of\_rent);

            dbms\_output.put\_line('-----------');

        END LOOP;

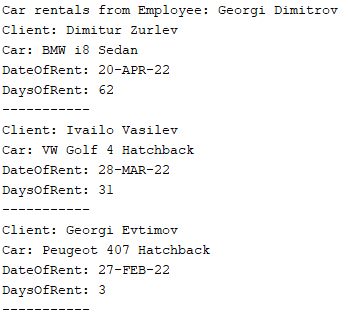
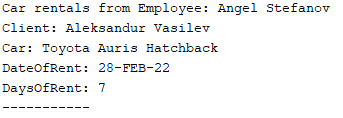
    END;

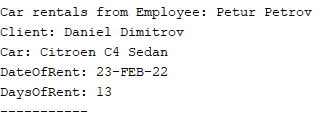
END;

* **Резултати**

set serveroutput on;

exec search\_car\_rentals\_by\_employee('&EMPLOYEE\_FULL\_NAME');





1. **Справка „Последните 10 заемания, подредени по дата“:**

* **Процедура**

create or **replace** PROCEDURE get\_latest\_rents

IS

BEGIN

    DECLARE

        CURSOR latest\_rents\_rows IS

        SELECT

            clients.full\_name     AS client\_full\_name,

            cars.brand as car\_brand,

            car\_models.model as car\_model,

            car\_types.type as car\_type,

            employees.full\_name   AS employee\_full\_name,

            date\_of\_rent,

            days\_of\_rent

        FROM

            rents

            INNER JOIN cars ON cars.id = car\_id

            INNER JOIN car\_models ON cars.model\_id = car\_models.id

            INNER JOIN car\_types ON cars.type\_id = car\_types.id

            INNER JOIN clients ON clients.id = client\_id

            INNER JOIN employees ON employees.id = employee\_id

        WHERE ROWNUM <= 10

        ORDER BY days\_of\_rent DESC;

    BEGIN

        dbms\_output.put\_line('Latest rents: ');

        FOR rent IN latest\_rents\_rows LOOP

            dbms\_output.put\_line('Client: ' || rent.client\_full\_name);

            dbms\_output.put\_line('Employee: ' || rent.employee\_full\_name);

            dbms\_output.put\_line('Car: ' || rent.car\_brand || ' ' || rent.car\_model || ' ' || rent.car\_type);

            dbms\_output.put\_line('DateOfRent: ' || rent.date\_of\_rent);

            dbms\_output.put\_line('DaysOfRent: ' || rent.days\_of\_rent);

            dbms\_output.put\_line('-----------');

        END LOOP;

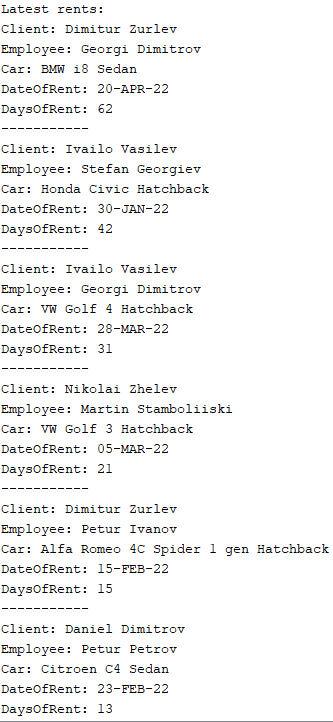
    END;

END;

* **Резултати**

set serveroutput on;

exec get\_latest\_rents;



1. **Справка „Наети автомобили от клиент,подредени по дата“:**

* **Процедура**

create or **replace** PROCEDURE search\_rented\_cars\_from\_client (

    s\_client\_full\_name clients.full\_name%TYPE

)

IS

BEGIN

    DECLARE

        CURSOR cars\_rented\_rows IS

        SELECT

            cars.brand as car\_brand,

            car\_models.model as car\_model,

            car\_types.type as car\_type,

            date\_of\_rent,

            days\_of\_rent

        FROM

            rents

            INNER JOIN cars ON cars.id = car\_id

            INNER JOIN car\_models ON cars.model\_id = car\_models.id

            INNER JOIN car\_types ON cars.type\_id = car\_types.id

            INNER JOIN clients ON clients.id = client\_id

            INNER JOIN employees ON employees.id = employee\_id

        WHERE

**upper**(clients.full\_name) = **upper**(s\_client\_full\_name)

        ORDER BY date\_of\_rent ASC;

    BEGIN

        dbms\_output.put\_line('Cars rented by: ' || s\_client\_full\_name);

        FOR car IN cars\_rented\_rows LOOP

            dbms\_output.put\_line('Car: ' || car.car\_brand || ' ' || car.car\_model || ' ' || car.car\_type);

            dbms\_output.put\_line('DateOfRent: ' || car.date\_of\_rent);

            dbms\_output.put\_line('DaysOfRent: ' || car.days\_of\_rent);

            dbms\_output.put\_line('-----------');

        END LOOP;

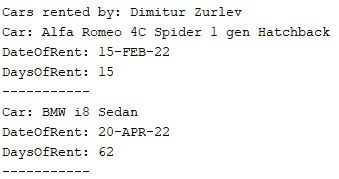
    END;

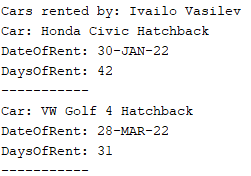
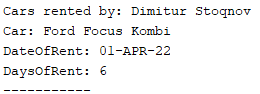
END;

* **Резултати**

set serveroutput on;

exec search\_rented\_cars\_from\_client('&CLIENT\_FULL\_NAME');





1. **Справка „Отдаване на автомобили за период, подредени по клиенти“:**

* **Процедура**

CREATE OR **REPLACE** PROCEDURE get\_rented\_cars\_in\_period (

    from\_period\_date date,

    to\_period\_date date

) IS

BEGIN

    DECLARE

        CURSOR latest\_rents\_rows IS

        SELECT

            clients.full\_name     AS client\_full\_name,

            cars.brand            AS car\_brand,

            car\_models.model      AS car\_model,

            car\_types.type        AS car\_type,

            employees.full\_name   AS employee\_full\_name,

            date\_of\_rent,

            days\_of\_rent

        FROM

            rents

            INNER JOIN cars ON cars.id = car\_id

            INNER JOIN car\_models ON cars.model\_id = car\_models.id

            INNER JOIN car\_types ON cars.type\_id = car\_types.id

            INNER JOIN clients ON clients.id = client\_id

            INNER JOIN employees ON employees.id = employee\_id

        WHERE

            date\_of\_rent BETWEEN from\_period\_date AND to\_period\_date

        ORDER BY

            clients.full\_name DESC;

    BEGIN

        dbms\_output.put\_line('Rents between: ' || from\_period\_date || ' and ' || to\_period\_date);

        FOR rent IN latest\_rents\_rows LOOP

            dbms\_output.put\_line('Client: ' || rent.client\_full\_name);

            dbms\_output.put\_line('Employee: ' || rent.employee\_full\_name);

            dbms\_output.put\_line('Car: ' || rent.car\_brand || ' ' || rent.car\_model || ' ' || rent.car\_type);

            dbms\_output.put\_line('DateOfRent: ' || rent.date\_of\_rent);

            dbms\_output.put\_line('DaysOfRent: ' || rent.days\_of\_rent);

            dbms\_output.put\_line('-----------');

        END LOOP;END;END;

* **Резултати**

set serveroutput on;

exec search\_rented\_cars\_in\_period(TO\_DATE('&FROM\_DATE', 'DD/MM/YYYY'), TO\_DATE('&TO\_DATE', 'DD/MM/YYYY'));

