**Технически университет – Варна**

Факултет: ФИТА

Катедра: Софтуерни и интернет технологии (СИТ)

Специалност: СИТ

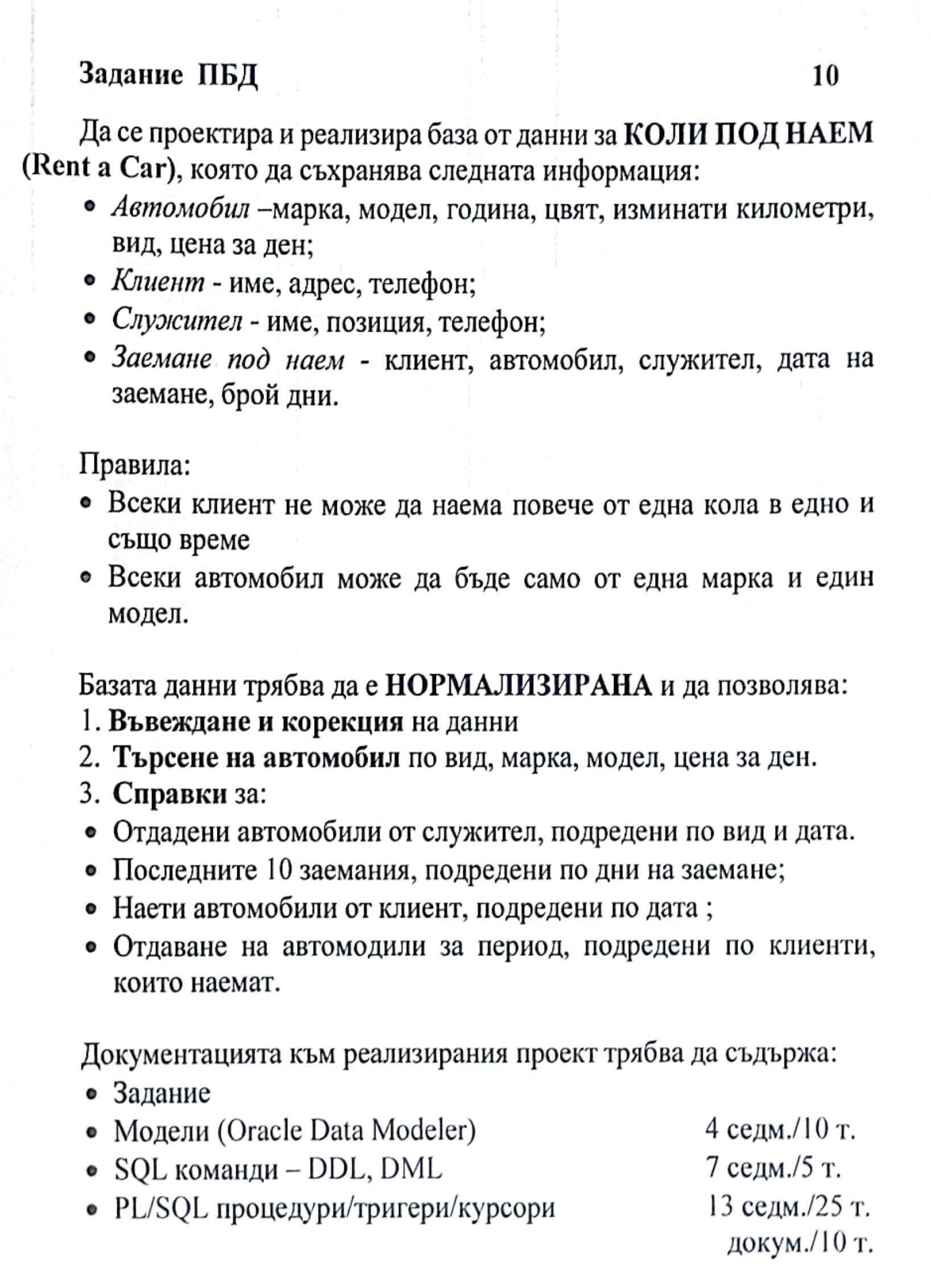
Тема на проекта:

Rent a Car, вариант 10

Изготвил:

Име: Теодор Николаев Николов

Факултетен номер: 20621504



Релационен модел

Diagram

Description automatically generated

Логически модел

Diagram

Description automatically generated

DML&DDL

Създаване на таблица: brand

- Съхраняваща марката на автомобила

CREATE TABLE brand (

id\_brand INTEGER NOT NULL,

brand VARCHAR2(20),

PRIMARY KEY ( id\_brand )

);

Добавяне на елементи в таблицата:

INSERT INTO BRAND(ID\_BRAND,BRAND) VALUES (1,'Ауди');

INSERT INTO BRAND(ID\_BRAND,BRAND) VALUES (2,'Опел');

INSERT INTO BRAND(ID\_BRAND,BRAND) VALUES (3,'БМВ');

INSERT INTO BRAND(ID\_BRAND,BRAND) VALUES (4,'Фолксваген');

INSERT INTO BRAND(ID\_BRAND,BRAND) VALUES (5,'Хонда');

INSERT INTO BRAND(ID\_BRAND,BRAND) VALUES (6,'Мерцедес');

INSERT INTO BRAND(ID\_BRAND,BRAND) VALUES (7,'Инфинити');

Актуализиране на елемент от таблицата:

UPDATE BRAND

SET BRAND='Тесла'

WHERE ID\_BRAND=7;

Изтриване на елемент от таблицата:

DELETE FROM BRAND WHERE ID\_BRAND=7;

Създаване на таблица: model

- Съхраняваща модела на автомобила, като всеки модел има точно определена марка.

CREATE TABLE model (

id\_model INTEGER NOT NULL,

model VARCHAR2(20),

brand\_id INTEGER NOT NULL,

PRIMARY KEY ( id\_model ),

CONSTRAINT model\_brand\_fk FOREIGN KEY ( brand\_id ) REFERENCES brand ( id\_brand )

);

Добавяне на елементи в таблицата:

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (1,'A5',1);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (2,'A7',1);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (3,'Q6',1);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (4,'Q7',1);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (5,'S8',1);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (6,'Вектра',2);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (7,'Корса',2);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (8,'Астра',2);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (9,'X6',3);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (10,'M6',3);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (11,'6',3);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (12,'Голф 7',4);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (13,'Тигуан',4);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (14,'Туарег',4);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (15,'Одисей',5);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (16,'Пилот',5);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (17,'B 220',6);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (18,'C 350',6);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (19,'E 400',6);

INSERT INTO MODEL(ID\_MODEL,MODEL,BRAND\_ID) VALUES (20,'G 320',6);

Актуализиране на елемент от таблицата:

UPDATE MODEL

SET MODEL='А3'

WHERE ID\_MODEL=1;

Изтриване на елемент от таблицата:

DELETE FROM MODEL WHERE ID\_MODEL=19;

Създаване на таблица: type

- Съхраняваща вида на автомобила

CREATE TABLE type (

id\_type INTEGER NOT NULL,

type VARCHAR2(20),

PRIMARY KEY ( id\_type )

);

Добавяне на елементи в таблицата:

INSERT INTO TYPE(ID\_TYPE,TYPE) VALUES (1,'Седан');

INSERT INTO TYPE(ID\_TYPE,TYPE) VALUES (2,'Хечбек');

INSERT INTO TYPE(ID\_TYPE,TYPE) VALUES (3,'Комби');

INSERT INTO TYPE(ID\_TYPE,TYPE) VALUES (4,'Джип');

INSERT INTO TYPE(ID\_TYPE,TYPE) VALUES (5,'Камион');

Актуализиране на елемент от таблицата:

UPDATE TYPE

SET type='Автобус'

WHERE ID\_TYPE=5;

Изтриване на елемент от таблицата:

DELETE FROM TYPE WHERE ID\_TYPE=5;

Създаване на таблица: color

- Съхраняваща цвета на автомобила

CREATE TABLE color (

id\_color INTEGER NOT NULL,

color VARCHAR2(20),

PRIMARY KEY ( id\_color )

);

Добавяне на елементи в таблицата:

INSERT INTO COLOR(ID\_COLOR,COLOR) VALUES(1,'бял');

INSERT INTO COLOR(ID\_COLOR,COLOR) VALUES(2,'черен');

INSERT INTO COLOR(ID\_COLOR,COLOR) VALUES(3,'син');

INSERT INTO COLOR(ID\_COLOR,COLOR) VALUES(4,'червен');

INSERT INTO COLOR(ID\_COLOR,COLOR) VALUES(5,'сив');

INSERT INTO COLOR(ID\_COLOR,COLOR) VALUES(6,'жълт');

Актуализиране на елемент от таблицата:

UPDATE COLOR

SET COLOR='виолетов'

WHERE ID\_COLOR=6;

Изтриване на елемент от таблицата:

DELETE FROM COLOR WHERE ID\_COLOR=6;

Създаване на таблица: car

- Съхраняваща автомобили, които имат определен модел, марка и цвят.

CREATE TABLE car (

id\_car INTEGER NOT NULL,

model\_id INTEGER NOT NULL,

type\_id INTEGER NOT NULL,

year INTEGER,

color\_id INTEGER NOT NULL,

traveled INTEGER,

daily\_price INTEGER,

PRIMARY KEY ( id\_car ),

CONSTRAINT car\_color\_fk FOREIGN KEY ( color\_id ) REFERENCES color ( id\_color ),

CONSTRAINT car\_model\_fk FOREIGN KEY ( model\_id ) REFERENCES model ( id\_model ),

CONSTRAINT car\_type\_fk FOREIGN KEY ( type\_id ) REFERENCES type ( id\_type )

);

Добавяне на елементи в таблицата:

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (1,1,1,2,120,2020,40000);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (2,1,1,1,55,2021,30200);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (3,4,1,4,155,2016,80000);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (4,3,2,7,30,2018,56923);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (5,2,2,8,120,2018,66534);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (6,4,4,13,135,2022,12000);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (7,4,3,9,145,2020,34523);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (8,4,5,16,130,2017,55423);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (9,3,4,12,95,2021,22123);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (10,1,3,10,185,2022,10000);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (11,4,6,20,210,2014,60542);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (12,2,6,18,175,2021,13243);

INSERT INTO CAR(ID\_CAR,TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES (13,1,2,6,135,2016,120345);

Актуализиране на елемент от таблицата:

UPDATE CAR

SET DAILY\_PRICE='100'

WHERE ID\_CAR=1;

Изтриване на елемент от таблицата:

DELETE FROM CAR WHERE ID\_CAR=8;

Създаване на таблица: client

- Съхраняваща клиенти.

CREATE TABLE client (

id\_client INTEGER NOT NULL,

name VARCHAR2(45),

address VARCHAR2(75),

telephone INTEGER,

PRIMARY KEY ( id\_client )

);

Добавяне на елементи в таблицата:

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (1,'Валентин Петров', 'София, ул.Васил Априлов №31, бл.6, ап.12', '0881015691');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (2,'Анастасия Яворова', 'София, ул.Алена Китка №12, бл.2, ап.78', '0898976431');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (3,'Иван Пенев', 'София, ул.6-ти септември №1, бл.5, ап.1', '0881812476');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (4,'Илия Илиев', 'Пловдив, ул.Иван Вазов №5, бл.3, ап.25', '0884759780');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (5,'Ангел Иванов', 'София, ул.Искър №67, бл.43, ап.22', '0891547689');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (6,'Калоян Герганов', 'София, ул.Искър №67, бл.43, ап.20', '0881043657');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (7,'Китка Узонова', 'Видин, ул.Княз Александър №10, бл.5, ап.97', '0885536981');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (8,'Калина Ивайлова', 'София, ул.Братя Миладинови №123, бл.45, ап.123', '0890123546');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (9,'Софи Маринова', 'София, ул.Баба Тонка №1, бл.12, ап.1', '0884618416');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (10,'Георги Петканов', 'Варна, ул.Струга №31, бл.41, ап.21', '0891436795');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (11,'Тодор Попов', 'София, ул.Батовска №12, бл.1, ап.112', '0889871439');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (12,'Светла Иванова', 'София, ул.Малка Поляна №43, бл.10, ап.15', '0893549653');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (13,'Георги Иванов', 'София, ул.Опълченска №4, бл.55, ап.125', '0881122984');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (14,'Иван Ивайлов', 'София, ул.Солунска №5, бл.11, ап.33', '0882785313');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (15,'Анита Георгиева', 'Пловдив, ул.Проектанска №11, бл.20, ап.78', '0881283500');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (16,'Александър Александров', 'София, ул.Китка №2, бл.2, ап.23', '0890874513');

INSERT INTO CLIENT (ID\_CLIENT, NAME, ADDRESS, TELEPHONE) VALUES (17,'Антония Александрова', 'София, ул.Китка №2, бл.2, ап.23', '0891234552');

Актуализиране на елемент от таблицата:

UPDATE CLIENT

SET NAME='Китка Петрова',ADDRESS='Враца, ул.Изгрев №5, бл.18, ап.22'

WHERE ID\_CLIENT=7;

Изтриване на елемент от таблицата:

DELETE FROM CLIENT WHERE ID\_CLIENT=6;

Създаване на таблица: position

- Съхраняваща позициите на служителите.

CREATE TABLE position (

id\_position INTEGER NOT NULL,

position VARCHAR2(40),

PRIMARY KEY ( id\_position )

);

Добавяне на елементи в таблицата:

INSERT INTO POSITION(ID\_POSITION,POSITION) VALUES (1,'Продавач-консултант');

INSERT INTO POSITION(ID\_POSITION,POSITION) VALUES (2,'Офис-сътрудник');

INSERT INTO POSITION(ID\_POSITION,POSITION) VALUES (3,'Счетоводител');

INSERT INTO POSITION(ID\_POSITION,POSITION) VALUES (4,'Юрист');

INSERT INTO POSITION(ID\_POSITION,POSITION) VALUES (5,'Администратор');

INSERT INTO POSITION(ID\_POSITION,POSITION) VALUES (6,'Механик');

INSERT INTO POSITION(ID\_POSITION,POSITION) VALUES (7,'Хигиенист');

INSERT INTO POSITION(ID\_POSITION,POSITION) VALUES (8,'Общ работник');

Актуализиране на елемент от таблицата:

UPDATE POSITION

SET POSITION='Пазач'

WHERE ID\_POSITION=8;

Изтриване на елемент от таблицата:

DELETE FROM POSITION WHERE ID\_POSITION=8;

Създаване на таблица: employees

- Съхраняваща служителите.

CREATE TABLE employees (

id\_employee INTEGER NOT NULL,

name VARCHAR2(45),

position\_id INTEGER NOT NULL,

telephone INTEGER,

PRIMARY KEY ( id\_employee ),

CONSTRAINT employees\_position\_fk FOREIGN KEY ( position\_id ) REFERENCES position ( id\_position )

);

Добавяне на елементи в таблицата:

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (1,'Андрей Едрев',2,'0885713647');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (2,'Моника Тодорова',2,'0884873641');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (3,'Васил Андреев',5,'0889510051');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (4,'Албена Божинова',5,'0889320032');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (5,'Мартин Николов',3,'0894325471');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (6,'Теодора Младенова',4,'0895364734');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (7,'Елена Ангелова',1,'0887418874');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (8,'Веселин Генчев',1,'0894624135');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (9,'Петър Иванов',1,'0884564128');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (10,'Тони Киров',6,'0884136547');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (11,'Ася Петрова',7,'0884725364');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (12,'Снежа Попова',7,'0885812387');

INSERT INTO EMPLOYEES(ID\_EMPLOYEE,NAME,POSITION\_ID,TELEPHONE) VALUES (13,'Гинка Стоева',7,'0894567142');

Актуализиране на елемент от таблицата:

UPDATE EMPLOYEES

SET TELEPHONE='0897641243'

WHERE ID\_EMPLOYEE=11;

Изтриване на елемент от таблицата:

DELETE FROM EMPLOYEES WHERE ID\_EMPLOYEE=13;

Създаване на таблица: rent

- Съхраняваща наемите на коли от клиенти.

CREATE TABLE rent (

id\_rent INTEGER NOT NULL,

client\_id INTEGER NOT NULL,

car\_id INTEGER NOT NULL,

employees\_id INTEGER NOT NULL,

rent\_date DATE,

days INTEGER,

PRIMARY KEY ( id\_rent ),

CONSTRAINT rent\_car\_fk FOREIGN KEY ( car\_id ) REFERENCES car ( id\_car ),

CONSTRAINT rent\_client\_fk FOREIGN KEY ( client\_id ) REFERENCES client ( id\_client ),

CONSTRAINT rent\_employees\_fk FOREIGN KEY ( employees\_id ) REFERENCES employees ( id\_employee )

);

Добавяне на елементи в таблицата:

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (1,5,3,8,DATE '2021-02-05',7);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (2,3,4,4,DATE '2021-03-11',10);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (3,2,5,3,DATE '2021-03-15',8);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (4,4,10,7,DATE '2021-05-03',15);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (5,10,11,3,DATE '2021-05-18',7);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (6,14,6,7,DATE '2021-05-25',5);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (7,8,12,9,DATE '2021-06-14',20);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (8,15,7,4,DATE '2021-07-08',31);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (9,17,6,8,DATE '2021-07-13',12);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (10,9,3,3,DATE '2021-11-21',5);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (11,1,1,9,DATE '2021-12-24',1);

INSERT INTO RENT(ID\_RENT,CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES (12,4,4,3,DATE '2021-12-30',5);

Актуализиране на елемент от таблицата:

UPDATE RENT

SET CAR\_ID=2

WHERE ID\_RENT=10;

Изтриване на елемент от таблицата:

DELETE FROM RENT WHERE ID\_RENT=6;

Търсения

**Търсене на автомобил по вид:**

SELECT type, brand, model, daily\_price

FROM CAR C

JOIN TYPE ON TYPE.id\_type=C.TYPE\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

WHERE TYPE\_ID LIKE '&type\_id'

ORDER BY daily\_price;

**Търсене на автомобил по марка:**

SELECT type, brand, model, daily\_price

FROM CAR C

JOIN TYPE ON TYPE.id\_type=C.TYPE\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

WHERE BRAND\_ID LIKE '&brand\_id'

ORDER BY daily\_price;

**Търсене на автомобил по модел:**

SELECT type, brand, model, daily\_price

FROM CAR C

JOIN TYPE ON TYPE.id\_type=C.TYPE\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

WHERE MODEL\_ID LIKE '&model\_id'

ORDER BY daily\_price;

**Търсене на автомобил по цена за ден:**

SELECT type, brand, model, daily\_price

FROM CAR C

JOIN TYPE ON TYPE.id\_type=C.TYPE\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

WHERE daily\_price LIKE '&price'

ORDER BY daily\_price;

Справки

**Отдадени автомобили от служител, подредени по вид и дата:**

SELECT CLIENT.ID\_CLIENT,EMPLOYEES.NAME AS emplyoee,TYPE,BRAND,MODEL,COLOR,TRAVELED,CLIENT.NAME AS client,RENT\_DATE

FROM CAR C

JOIN RENT ON RENT.CAR\_ID=C.ID\_CAR

JOIN TYPE ON TYPE.ID\_TYPE=C.TYPE\_ID

JOIN COLOR ON COLOR.ID\_COLOR=C.COLOR\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

JOIN CLIENT ON CLIENT.ID\_CLIENT=RENT.CLIENT\_ID

JOIN EMPLOYEES ON EMPLOYEES.ID\_EMPLOYEE=RENT.EMPLOYEES\_ID

WHERE EMPLOYEES\_ID LIKE '&employee\_id'

ORDER BY C.TYPE\_ID,RENT\_date;

**Последните 10 заемания, подредени по брой дни на заемане:**

SELECT \* FROM

(SELECT CLIENT.NAME AS client,EMPLOYEES.NAME AS employee,POSITION,BRAND,MODEL,RENT\_DATE,DAYS,DAILY\_PRICE\*DAYS AS total FROM RENT R

JOIN CLIENT ON CLIENT.ID\_CLIENT=R.CLIENT\_ID

JOIN CAR ON CAR.ID\_CAR=R.CAR\_ID

JOIN MODEL M ON M.id\_model=CAR.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

JOIN EMPLOYEES ON EMPLOYEES.ID\_EMPLOYEE=R.EMPLOYEES\_ID

JOIN POSITION ON POSITION.ID\_POSITION=EMPLOYEES.POSITION\_ID

ORDER BY DAYS DESC)

WHERE ROWNUM<=10;

**Наети автомобили от клиент, подредени по дата:**

SELECT CLIENT.NAME AS client,BRAND,MODEL,EMPLOYEES.NAME AS employee,RENT\_DATE,DAILY\_PRICE\*DAYS AS total FROM RENT R

JOIN CLIENT ON CLIENT.ID\_CLIENT=R.CLIENT\_ID

JOIN CAR ON CAR.ID\_CAR=R.CAR\_ID

JOIN MODEL M ON M.id\_model=CAR.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

JOIN EMPLOYEES ON EMPLOYEES.ID\_EMPLOYEE=R.EMPLOYEES\_ID

WHERE CLIENT\_ID LIKE '&client\_id'

ORDER BY RENT\_DATE;

**Отдаване на автомобили за период, подредени по клиенти\_id:**

SELECT CLIENT.NAME AS client,BRAND,MODEL,EMPLOYEES.NAME AS employee,RENT\_DATE,DAILY\_PRICE\*DAYS AS total FROM RENT R

JOIN CLIENT ON CLIENT.ID\_CLIENT=R.CLIENT\_ID

JOIN CAR ON CAR.ID\_CAR=R.CAR\_ID

JOIN MODEL M ON M.id\_model=CAR.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

JOIN EMPLOYEES ON EMPLOYEES.ID\_EMPLOYEE=R.EMPLOYEES\_ID

WHERE RENT\_DATE BETWEEN '&Начална\_дата' and '&Крайна\_дата'

ORDER BY CLIENT\_ID;

PL/SQL

**Тригери**

Създаване на тригер за автоматично въвеждане на ID, таблица Brand

CREATE SEQUENCE brand\_sequence;

CREATE OR REPLACE TRIGGER brand\_on\_insert

BEFORE INSERT ON brand

FOR EACH ROW

BEGIN

SELECT brand\_sequence.nextval

INTO :new.id\_brand

FROM dual;

END;

Създаване на тригер за автоматично въвеждане на ID, таблица Car

CREATE SEQUENCE car\_sequence;

CREATE OR REPLACE TRIGGER car\_on\_insert

BEFORE INSERT ON car

FOR EACH ROW

BEGIN

SELECT car\_sequence.nextval

INTO :new.id\_car

FROM dual;

END;

Създаване на тригер за автоматично въвеждане на ID, таблица Client

CREATE SEQUENCE client\_sequence;

CREATE OR REPLACE TRIGGER client\_on\_insert

BEFORE INSERT ON client

FOR EACH ROW

BEGIN

SELECT client\_sequence.nextval

INTO :new.id\_client

FROM dual;

END;

Създаване на тригер за автоматично въвеждане на ID, таблица Color

CREATE SEQUENCE color\_sequence;

CREATE OR REPLACE TRIGGER color\_on\_insert

BEFORE INSERT ON color

FOR EACH ROW

BEGIN

SELECT color\_sequence.nextval

INTO :new.id\_color

FROM dual;

END;

Създаване на тригер за автоматично въвеждане на ID, таблица Employees

CREATE SEQUENCE employees\_sequence;

CREATE OR REPLACE TRIGGER employees\_on\_insert

BEFORE INSERT ON employees

FOR EACH ROW

BEGIN

SELECT employees\_sequence.nextval

INTO :new.id\_employee

FROM dual;

END;

Създаване на тригер за автоматично въвеждане на ID, таблица Model

CREATE SEQUENCE model\_sequence;

CREATE OR REPLACE TRIGGER model\_on\_insert

BEFORE INSERT ON model

FOR EACH ROW

BEGIN

SELECT model\_sequence.nextval

INTO :new.id\_model

FROM dual;

END;

Създаване на тригер за автоматично въвеждане на ID, таблица Position

CREATE SEQUENCE position\_sequence;

CREATE OR REPLACE TRIGGER position\_on\_insert

BEFORE INSERT ON position

FOR EACH ROW

BEGIN

SELECT position\_sequence.nextval

INTO :new.id\_position

FROM dual;

END;

Създаване на тригер за автоматично въвеждане на ID, таблица Rent

CREATE SEQUENCE rent\_sequence;

CREATE OR REPLACE TRIGGER rent\_on\_insert

BEFORE INSERT ON rent

FOR EACH ROW

BEGIN

SELECT rent\_sequence.nextval

INTO :new.id\_rent

FROM dual;

END;

Създаване на тригер за автоматично въвеждане на ID, таблица Type

CREATE SEQUENCE type\_sequence;

CREATE OR REPLACE TRIGGER type\_on\_insert

BEFORE INSERT ON type

FOR EACH ROW

BEGIN

SELECT type\_sequence.nextval

INTO :new.id\_type

FROM dual;

END;

**Процедури**

Процедура за добавяне на елементи в таблица Brand

CREATE OR REPLACE PROCEDURE InsertBrand(input\_brand IN brand.brand%TYPE) IS

BEGIN

INSERT INTO BRAND(brand) VALUES(input\_brand);

Graphical user interface, application

Description automatically generatedEND;

exec InsertBrand('&brand');

Graphical user interface, application

Description automatically generated

Процедура за актуализиране на елементи в таблица Brand

CREATE OR REPLACE PROCEDURE UpdateBrand(input\_id IN brand.id\_brand%TYPE, input\_brand IN brand.brand%TYPE) IS

BEGIN

UPDATE BRAND SET brand = input\_brand WHERE id\_brand = input\_id;

END;

exec UpdateBrand('&brand\_id', '&new\_brand');

Процедура за изтриване на елементи по ид в таблица Brand

CREATE OR REPLACE PROCEDURE DeleteBrand(input\_id IN brand.id\_brand%TYPE) IS

BEGIN

DELETE FROM BRAND WHERE id\_brand = input\_id;

END;

Graphical user interface, application

Description automatically generatedexec DeleteBrand('&brand\_id');

Graphical user interface, application

Description automatically generated

Процедура за добавяне на елементи в таблица Model

CREATE OR REPLACE PROCEDURE InsertModel(input\_model IN model.model%TYPE, input\_brand IN brand.brand%TYPE)

IS

id INTEGER;

BEGIN

SELECT id\_brand INTO id FROM BRAND WHERE brand=input\_brand;

INSERT INTO MODEL(model,brand\_id) VALUES(input\_model,id);

END;

exec InsertModel('&model','&brand');

Процедура за актуализиране на елементи в таблица Model

CREATE OR REPLACE PROCEDURE UpdateModel(input\_id IN model.id\_model%TYPE, input\_model IN model.model%TYPE) IS

BEGIN

UPDATE MODEL SET model = input\_model WHERE id\_model = input\_id;

END;

exec UpdateModel('&model\_id', '&new\_model');

Процедура за изтриване на елементи по ид в таблица Model

CREATE OR REPLACE PROCEDURE DeleteModel(input\_id IN model.id\_model%TYPE) IS

BEGIN

DELETE FROM MODEL WHERE id\_model = input\_id;

END;

exec DeleteModel('&model\_id');

Процедура за добавяне на елементи в таблица Type

CREATE OR REPLACE PROCEDURE InsertType(input\_type IN type.type%TYPE) IS

BEGIN

INSERT INTO TYPE(type) VALUES(input\_type);

END;

exec InsertType('&type');

Graphical user interface

Description automatically generated with low confidenceGraphical user interface, application

Description automatically generated

Процедура за актуализиране на елементи в таблица Type

CREATE OR REPLACE PROCEDURE UpdateType(input\_id IN type.id\_type%TYPE, input\_type IN type.type%TYPE) IS

BEGIN

UPDATE TYPE SET type = input\_type WHERE id\_type = input\_id;

END;

exec UpdateType('&type\_id', '&new\_type');

Graphical user interface

Description automatically generated Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated with medium confidence

Процедура за изтриване на елементи по ид в таблица Type

CREATE OR REPLACE PROCEDURE DeleteType(input\_id IN model.id\_model%TYPE) IS

BEGIN

DELETE FROM TYPE WHERE id\_type = input\_id;

END;

exec DeleteType('&type\_id');

Процедура за добавяне на елементи в таблица Color

CREATE OR REPLACE PROCEDURE InsertColor(input\_color IN color.color%TYPE) IS

Graphical user interface, application

Description automatically generatedBEGIN

INSERT INTO COLOR(color) VALUES(input\_color);

END;

exec InsertColor('&color');

Процедура за актуализиране на елементи в таблица Color

CREATE OR REPLACE PROCEDURE UpdateColor(input\_id IN INTEGER, input\_color IN VARCHAR2) IS

BEGIN

UPDATE COLOR SET color = input\_color WHERE id\_color = input\_id;

END;

exec UpdateColor('&color\_id', '&new\_color');

Graphical user interface, application

Description automatically generated Graphical user interface, application, website

Description automatically generated Graphical user interface

Description automatically generated

Процедура за изтриване на елементи по ид в таблица Color

CREATE OR REPLACE PROCEDURE DeleteColor(input\_id IN INTEGER) IS

BEGIN

DELETE FROM COLOR WHERE id\_color = input\_id;

END;

exec DeleteColor('&color\_id');s

Graphical user interface, application

Description automatically generated Graphical user interface

Description automatically generated with medium confidence

Процедура за добавяне на елементи в таблица Car

CREATE OR REPLACE PROCEDURE InsertCar(input\_type IN type.type%TYPE, input\_color IN color.color%TYPE, input\_model IN model.model%TYPE, input\_price IN car.daily\_price%TYPE, input\_year IN car.year%TYPE, input\_traveled IN car.traveled%TYPE)

IS

id\_type INTEGER;

id\_color INTEGER;

id\_model INTEGER;

BEGIN

SELECT id\_type INTO id\_type FROM TYPE WHERE type=input\_type;

SELECT id\_color INTO id\_color FROM COLOR WHERE color=input\_color;

SELECT id\_model INTO id\_model FROM MODEL WHERE model=input\_model;

INSERT INTO CAR(TYPE\_ID,COLOR\_ID,MODEL\_ID,DAILY\_PRICE,YEAR,TRAVELED) VALUES(id\_type,id\_color,id\_model,input\_price,input\_year,input\_traveled);

END;

exec InsertCar('&type','&color','&model','&daily\_price','&year','&traveled');

Процедура за актуализиране на елементи в таблица Car

CREATE OR REPLACE PROCEDURE UpdateCar(input\_id IN INTEGER, input\_traveled IN INTEGER) IS

BEGIN

UPDATE CAR SET traveled = input\_traveled WHERE id\_car = input\_id;

END;

exec UpdateCar('&car\_id', '&new\_traveled');

Процедура за изтриване на елементи по ид в таблица Car

CREATE OR REPLACE PROCEDURE DeleteCar(input\_id IN INTEGER) IS

BEGIN

DELETE FROM CAR WHERE id\_car = input\_id;

END;

exec DeleteCar('&car\_id');

Процедура за добавяне на елементи в таблица Positon

CREATE OR REPLACE PROCEDURE InsertPosition(input\_position IN position.position%TYPE) IS

BEGIN

INSERT INTO POSITION(position) VALUES(input\_position);

END;

exec InsertPosition('&position');

Процедура за актуализиране на елементи в таблица Positon

CREATE OR REPLACE PROCEDURE UpdatePosition(input\_id IN INTEGER, input\_position IN VARCHAR2) IS

BEGIN

UPDATE POSITION SET position = input\_position WHERE id\_position = input\_id;

END;

exec UpdatePosition('&position\_id', '&new\_position');

A screenshot of a computer

Description automatically generated with medium confidence Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated A screenshot of a computer

Description automatically generated with medium confidence

Процедура за изтриване на елементи по ид в таблица Positon

CREATE OR REPLACE PROCEDURE DeletePosition(input\_id IN INTEGER) IS

BEGIN

DELETE FROM POSITION WHERE id\_position = input\_id;

END;

exec DeletePosition('&position\_id');

Процедура за добавяне на елементи в таблица Employee

CREATE OR REPLACE PROCEDURE InsertEmployee(input\_name IN employees.name%TYPE, input\_position IN position.position%TYPE, input\_telephone IN employees.telephone%TYPE)

IS

id INTEGER;

BEGIN

SELECT id\_position INTO id FROM POSITION WHERE position=input\_position;

INSERT INTO EMPLOYEES(name,position\_id,telephone) VALUES(input\_name,id,input\_telephone);

END;

exec InsertEmployee('&name','&position','&telephone');

Процедура за актуализиране на елементи в таблица Employee

CREATE OR REPLACE PROCEDURE UpdateEmployee(input\_id IN INTEGER, input\_telephone IN INTEGER) IS

BEGIN

UPDATE EMPLOYEES SET telephone = input\_telephone WHERE id\_employee = input\_id;

END;

exec UpdateEmployee('&employee', '&new\_telephone');

Процедура за изтриване на елементи по ид в таблица Employee

CREATE OR REPLACE PROCEDURE DeleteEmployee(input\_id IN INTEGER) IS

BEGIN

DELETE FROM EMPLOYEES WHERE id\_employee = input\_id;

END;

exec DeleteEmployee('&employee\_id');

Процедура за добавяне на елементи в таблица Client

CREATE OR REPLACE PROCEDURE InsertClient(input\_name IN client.name%TYPE, input\_address IN client.address%TYPE, input\_telephone IN client.telephone%TYPE) IS

BEGIN

INSERT INTO CLIENT(name,address,telephone) VALUES(input\_name,input\_address,input\_telephone);

END;

exec InsertClient('&name','&address','&telephone');

Процедура за актуализиране на елементи в таблица Client

CREATE OR REPLACE PROCEDURE UpdateClient(input\_id IN INTEGER, input\_telephone IN INTEGER) IS

BEGIN

UPDATE CLIENT SET telephone = input\_telephone WHERE id\_client = input\_id;

END;

exec UpdateClient('&client\_id', '&new\_telephone');

Процедура за изтриване на елементи по ид в таблица Client

CREATE OR REPLACE PROCEDURE DeleteClient(input\_id IN INTEGER) IS

BEGIN

DELETE FROM CLIENT WHERE id\_client = input\_id;

END;

exec DeleteClient('&client\_id');

Процедура за добавяне на елементи в таблица Rent

-Проверка по наемане на дата дали колата е свободна.

CREATE OR REPLACE PROCEDURE InsertRent(input\_client IN client.name%TYPE, input\_car\_id IN rent.car\_id%TYPE, input\_employee IN employees.name%TYPE, input\_date IN rent.rent\_date%TYPE, input\_days IN rent.days%TYPE)

IS

id\_client INTEGER;

id\_employee INTEGER;

date\_old DATE;

days INTEGER;

max\_rows INTEGER;

flag BOOLEAN;

BEGIN

SELECT COUNT(input\_car\_id) INTO max\_rows FROM RENT WHERE car\_id=input\_car\_id;

SELECT id\_client INTO id\_client FROM CLIENT WHERE name=input\_client;

SELECT id\_employee INTO id\_employee FROM EMPLOYEES WHERE name=input\_employee;

flag:=true;

FOR counter IN 1..max\_rows loop

SELECT rent\_date INTO date\_old FROM(SELECT rownum rn, rent\_date FROM RENT WHERE car\_id=input\_car\_id ORDER BY id\_rent) WHERE rn=counter;

SELECT days INTO days FROM(SELECT rownum rn, days FROM RENT WHERE car\_id=input\_car\_id ORDER BY id\_rent) WHERE rn=counter;

IF date\_old + days > input\_date

THEN flag:=false;

END IF;

END LOOP;

IF flag

THEN INSERT INTO RENT(CLIENT\_ID,CAR\_ID,EMPLOYEES\_ID,RENT\_DATE,DAYS) VALUES(id\_client,input\_car\_id,id\_employee, input\_date,input\_days);

END IF;

END;

exec InsertRent('&client','&car\_id','&employee','&rent\_date','&days');

Процедура за актуализиране на елементи в таблица Rent

CREATE OR REPLACE PROCEDURE UpdateRent(input\_id IN INTEGER, input\_days IN INTEGER) IS

BEGIN

UPDATE RENT SET days = input\_days WHERE id\_rent = input\_id;

END;

exec UpdateRent('&rent\_id', '&new\_days');

Процедура за изтриване на елементи по ид в таблица Rent

CREATE OR REPLACE PROCEDURE DeleteRent(input\_id IN INTEGER) IS

BEGIN

DELETE FROM RENT WHERE id\_rent = input\_id;

END;

exec DeleteRent('&rent\_id');

Процедура за търсене на автомобил по вид:

CREATE OR REPLACE PROCEDURE SearchCarByType(input\_type IN VARCHAR2) IS

BEGIN

FOR v\_record IN ( SELECT type, brand, model, daily\_price

FROM CAR C

JOIN TYPE ON TYPE.id\_type=C.TYPE\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

WHERE type LIKE input\_type

ORDER BY daily\_price)

LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_record.type || ' ' ||v\_record.brand || ' ' || v\_record.model || '-цена за ден:' || v\_record.daily\_price);

END LOOP;

END;

exec SearchCarByType('&type\_id');

Graphical user interface, application

Description automatically generated Text

Description automatically generated

Процедура за търсене на автомобил по марка:

CREATE OR REPLACE PROCEDURE SearchCarByBrand(input\_brand IN VARCHAR2) IS

BEGIN

FOR v\_record IN ( SELECT type, brand, model, daily\_price

FROM CAR C

JOIN TYPE ON TYPE.id\_type=C.TYPE\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

WHERE brand LIKE input\_brand

ORDER BY daily\_price)

LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_record.type || ' ' ||v\_record.brand || ' ' || v\_record.model || '-цена за ден:' || v\_record.daily\_price);

END LOOP;

END;

exec SearchCarByBrand('&brand');

Graphical user interface, text, application

Description automatically generated Text

Description automatically generated

Процедура за търсене на автомобил по модел:

CREATE OR REPLACE PROCEDURE SearchCarByModel(input\_model IN INTEGER) IS

BEGIN

FOR v\_record IN ( SELECT type, brand, model, daily\_price

FROM CAR C

JOIN TYPE ON TYPE.id\_type=C.TYPE\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

WHERE model LIKE input\_model

ORDER BY daily\_price)

LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_record.type || ' ' ||v\_record.brand || ' ' || v\_record.model || '-цена за ден:' || v\_record.daily\_price);

END LOOP;

END;

exec SearchCarByModel('&model');

Graphical user interface, application

Description automatically generated Text

Description automatically generated with medium confidence

Процедура за търсене на автомобил по цена за ден:

CREATE OR REPLACE PROCEDURE SearchCarByDailyPrice(input\_daily\_price IN INTEGER) IS

BEGIN

FOR v\_record IN ( SELECT type, brand, model, daily\_price

FROM CAR C

JOIN TYPE ON TYPE.id\_type=C.TYPE\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

WHERE daily\_price LIKE input\_daily\_price

ORDER BY daily\_price)

LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_record.type || ' ' ||v\_record.brand || ' ' || v\_record.model || '-цена за ден:' || v\_record.daily\_price);

END LOOP;

END;

exec SearchCarByDailyPrice('&daily\_price');

Graphical user interface, text, application

Description automatically generated Text

Description automatically generated

Процедура за отдадени автомобили от служител, подредени по вид и дата:

CREATE OR REPLACE PROCEDURE SearchRentByEmployee(input\_employee\_name IN VARCHAR2)

IS

selected\_employee VARCHAR2(255);

BEGIN

selected\_employee:= '%'||input\_employee\_name||'%';

FOR v\_record IN ( SELECT EMPLOYEES.NAME AS emplyoee,TYPE,BRAND,MODEL,COLOR,TRAVELED,CLIENT.NAME AS client,RENT\_DATE

FROM CAR C

JOIN RENT ON RENT.CAR\_ID=C.ID\_CAR

JOIN TYPE ON TYPE.ID\_TYPE=C.TYPE\_ID

JOIN COLOR ON COLOR.ID\_COLOR=C.COLOR\_ID

JOIN MODEL M ON M.id\_model=C.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

JOIN CLIENT ON CLIENT.ID\_CLIENT=RENT.CLIENT\_ID

JOIN EMPLOYEES ON EMPLOYEES.ID\_EMPLOYEE=RENT.EMPLOYEES\_ID

WHERE EMPLOYEES.NAME LIKE selected\_employee

ORDER BY C.TYPE\_ID,RENT\_date)

LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_record.emplyoee || ': ' || v\_record.TYPE || '-' || v\_record.BRAND || '-' || v\_record.MODEL || '-' || v\_record.COLOR || ', изминати киометри:' || v\_record.TRAVELED || ', отдадена на: ' || v\_record.client || ' дата:' || v\_record.RENT\_DATE);

END LOOP;

END;

exec SearchRentByEmployee ('&employee');

A screenshot of a computer

Description automatically generated with medium confidenceGraphical user interface, text, application

Description automatically generated

Процедура за последните 10 заемания, подредени по брой дни на заемане:

CREATE OR REPLACE PROCEDURE LastTen IS

BEGIN

FOR v\_record IN ( SELECT \* FROM

(SELECT CLIENT.NAME AS client,EMPLOYEES.NAME AS employee,POSITION,BRAND,MODEL,RENT\_DATE,DAYS,DAILY\_PRICE\*DAYS AS total FROM RENT R

JOIN CLIENT ON CLIENT.ID\_CLIENT=R.CLIENT\_ID

JOIN CAR ON CAR.ID\_CAR=R.CAR\_ID

JOIN MODEL M ON M.id\_model=CAR.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

JOIN EMPLOYEES ON EMPLOYEES.ID\_EMPLOYEE=R.EMPLOYEES\_ID

JOIN POSITION ON POSITION.ID\_POSITION=EMPLOYEES.POSITION\_ID

ORDER BY DAYS DESC)

WHERE ROWNUM<=10)

LOOP

DBMS\_OUTPUT.PUT\_LINE('клиент: ' || v\_record.client || ' служител: ' || v\_record.employee || '-' || v\_record.POSITION || ' кола: ' || v\_record.BRAND || '-' || v\_record.MODEL || ' дата на наемане: ' || v\_record.RENT\_DATE || ' дни: ' || v\_record.DAYS || ' цена общо: ' || v\_record.total || 'лв.');

END LOOP;

END;

exec LastTen();

A picture containing text

Description automatically generated

Процедура за наети автомобили от клиент, подредени по дата:

CREATE OR REPLACE PROCEDURE SearchRentByClient (input\_client\_name IN VARCHAR2)

IS

selected\_client VARCHAR2(255);

BEGIN

selected\_client:= '%'||input\_client\_name||'%';

FOR v\_record IN ( SELECT CLIENT.NAME AS client,BRAND,MODEL,EMPLOYEES.NAME AS employee,RENT\_DATE,DAILY\_PRICE\*DAYS AS total FROM RENT R

JOIN CLIENT ON CLIENT.ID\_CLIENT=R.CLIENT\_ID

JOIN CAR ON CAR.ID\_CAR=R.CAR\_ID

JOIN MODEL M ON M.id\_model=CAR.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

JOIN EMPLOYEES ON EMPLOYEES.ID\_EMPLOYEE=R.EMPLOYEES\_ID

WHERE CLIENT.NAME LIKE selected\_client

ORDER BY RENT\_DATE)

LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_record.client || ': ' || v\_record.BRAND || '-' || v\_record.MODEL || ', служител: ' || v\_record.employee || ', дата:' || v\_record.RENT\_DATE || ', цена общо: ' || v\_record.total || 'лв.');

END LOOP;

END;

exec SearchRentByClient ('&client');

A screenshot of a computer

Description automatically generated with medium confidenceGraphical user interface, application

Description automatically generated

Процедура за отдаване на автомобили за период, подредени по клиенти\_id:

CREATE OR REPLACE PROCEDURE SearchRentBetweenDates(input\_from\_date IN DATE, input\_to\_date IN DATE)IS

BEGIN

FOR v\_record IN ( SELECT CLIENT.NAME AS client,BRAND,MODEL,EMPLOYEES.NAME AS employee,RENT\_DATE,DAILY\_PRICE\*DAYS AS total FROM RENT R

JOIN CLIENT ON CLIENT.ID\_CLIENT=R.CLIENT\_ID

JOIN CAR ON CAR.ID\_CAR=R.CAR\_ID

JOIN MODEL M ON M.id\_model=CAR.MODEL\_ID

JOIN BRAND ON BRAND.id\_brand=M.BRAND\_ID

JOIN EMPLOYEES ON EMPLOYEES.ID\_EMPLOYEE=R.EMPLOYEES\_ID

WHERE RENT\_DATE BETWEEN input\_from\_date and input\_to\_date

ORDER BY CLIENT\_ID)

LOOP

DBMS\_OUTPUT.PUT\_LINE(v\_record.client || ': ' || v\_record.BRAND || '-' || v\_record.MODEL || ', служител: ' || v\_record.employee || ', дата:' || v\_record.RENT\_DATE || ', цена общо: ' || v\_record.total || 'лв.');

END LOOP;

END;

exec SearchRentBetweenDates ('&from\_date','&to\_date');

Graphical user interface, application

Description automatically generated Graphical user interface, text, application

Description automatically generated A screenshot of a computer

Description automatically generated with medium confidence