**НОВ БЪЛГАРСКИ УНИВЕРСИТЕТ**

**INFM111 Програмиране с PL/SQL**

Тема: База данни за сервиз за автомобили

Изготвил: **Симеон Валериев Михайлов** Ръководител:

Факултетен номер: **F83533** **доц. д-р. Георги Тупаров**

Департамент: **Информатика**

Програма: **Информатика**

**Описание на темата**

В автосервиз има техници, клиенти и автомобили. Техниците имат определен час, в който да започнат работа върху автомобил. Един техник може да работи върху един или повече автомобили и върху един автомобил могат да работят един или повече техници.

**Структура на базата данни**

За осъществяването на базата данни са необходими 4 таблици – Technicians, Vehicles, Customers и Appointments.

Таблицата за техници ще съдържа име на техника, област на дейност и телефонен номер. Всички полета в таблицата са от тип символен низ като името и областта на дейност могат да съдържат до 50 символа, докато телефонният номер – до 10 символа.

Таблицата за автомобили ще съдържа регистрационен номер, марка и модел на превозното средство. Всички полета са от тип символен низ като марката и моделът могат да съдържат до 50 символа, докато регистрационният номер – до 8 символа.

Таблицата за клиенти ще съдържа име на клиента, адрес на клиента, телефонен номер и регистрационен номер на автомобила му. Всички полета са от тип символен низ като името може да съдържа до 50 символа, адресът до 100, телефонният номер до 10 и регистрационният номер на автомобила му – до 8 символа.

Таблицата за назначените часове, в които техниците трябва да започнат работа по автомобил ще съдържа телефонен номер на техник, регистрационен номер на автомобил и дата и час. Телефонният номер на техника и регистрационният номер на автомобила са от тип символен низ като телефонният номер може да съдържа до 10 символа, докато регистрационният номер – до 8 символа. Полето за дата и час е от тип timestamp.

DROP TABLE Appointments;

DROP TABLE Customers;

DROP TABLE Vehicles;

DROP TABLE Technicians;

**Таблица Technicians**

CREATE TABLE Technicians (

technician\_name VARCHAR2(50) NOT NULL,

technician\_field VARCHAR2(50) NOT NULL,

technician\_phone\_number VARCHAR2(10) NOT NULL,

CONSTRAINT technicians\_pk PRIMARY KEY (technician\_phone\_number)

);

*Примерни данни за таблица Technicians*

TECHNICIAN\_NAME TECHNICIAN\_FIELD TECHNICIAN\_PHONE\_NUMBER

Peter Parker Engine Mechanic 0887994106

Tony Stark Brakes Mechanic 0876479234

**Таблица Vehicles**

CREATE TABLE Vehicles (

vehicle\_license\_plate VARCHAR2(8) NOT NULL,

vehicle\_make VARCHAR2(50) NOT NULL,

vehicle\_model VARCHAR2(50) NOT NULL,

CONSTRAINT vehicles\_pk PRIMARY KEY (vehicle\_license\_plate)

);

*Примерни данни за таблица Vehicles*

VEHICLE\_LICENSE\_PLATE VEHICLE\_MAKE VEHICLE\_MODEL

CB2579KP Honda Accord

CA8360EC VW Golf

**Таблица Customers**

CREATE TABLE Customers (

customer\_name VARCHAR2(50) NOT NULL,

customer\_address VARCHAR2(100) NOT NULL,

customer\_phone\_number VARCHAR2(10) NOT NULL,

vehicle\_license\_plate VARCHAR2(8) NOT NULL,

CONSTRAINT customers\_pk PRIMARY KEY (customer\_phone\_number),

CONSTRAINT fk\_vehicles FOREIGN KEY (vehicle\_license\_plate) REFERENCES vehicles(vehicle\_license\_plate)

);

*Примерни данни за таблица Customers*

CUSTOMER\_NAME CUSTOMER\_ADDRESS CUSTOMER\_PHONE\_NUMBER VEHICLE\_LICENSE\_PLATE

Sofia Carson NYC, USA 0735709823 CA8360EC

Madison Beer Chicago, USA 0926396183 CB2579KP

**Таблица Appointments**

CREATE TABLE Appointments (

technician\_phone\_number VARCHAR2(10) NOT NULL,

vehicle\_license\_plate VARCHAR2(8) NOT NULL,

appointment\_timestamp TIMESTAMP NOT NULL,

CONSTRAINT appointments\_pk PRIMARY KEY (technician\_phone\_number, vehicle\_license\_plate),

CONSTRAINT fk\_technicians FOREIGN KEY (technician\_phone\_number) REFERENCES technicians(technician\_phone\_number),

CONSTRAINT fk\_vehicles2 FOREIGN KEY (vehicle\_license\_plate) REFERENCES vehicles(vehicle\_license\_plate)

);

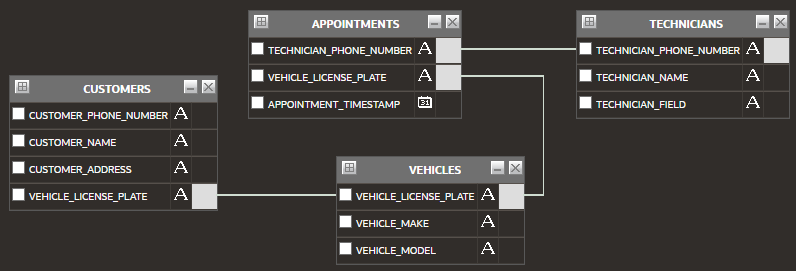
*Примерни данни за таблица Appointments*

TECHNICIAN\_PHONE\_NUMBER VEHICLE\_LICENSE\_PLATE APPOINTMENT\_TIMESTAMP

0887994106 CB2579KP 29-DEC-20 11.15.00.000000 AM

0876479234 CA8360EC 29-DEC-20 08.30.25.000000 AM

Диаграма на връзките между таблиците



**Задачи**

**Процедура за добавяне на нов техник:**

CREATE OR REPLACE PROCEDURE AddNewTechnician(i\_name IN technicians.technician\_name%TYPE, i\_field IN technicians.technician\_field%TYPE, i\_phone\_number IN technicians.technician\_phone\_number%TYPE)

AS

v\_technician\_exists NUMBER;

v\_phone\_number\_taken\_by\_customer NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_technician\_exists FROM technicians WHERE i\_phone\_number = technician\_phone\_number;

SELECT COUNT(\*) INTO v\_phone\_number\_taken\_by\_customer FROM customers WHERE i\_phone\_number = customer\_phone\_number;

IF v\_technician\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This phone number is already used by an existing technician.');

ELSIF v\_phone\_number\_taken\_by\_customer > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This phone number is already used by an existing customer.');

ELSE

INSERT INTO technicians(technician\_name, technician\_field, technician\_phone\_number) VALUES (i\_name, i\_field, i\_phone\_number);

DBMS\_OUTPUT.PUT\_LINE('Technician successfully added to the database.');

END IF;

END AddNewTechnician;

*Код за извикване:*

DECLARE

i\_name technicians.technician\_name%TYPE := 'Peter Parker';

i\_field technicians.technician\_field%TYPE := 'Engine Mechanic';

i\_phone\_number technicians.technician\_phone\_number%TYPE := '1234567890';

BEGIN

vehicle\_service.AddNewTechnician(i\_name, i\_field, i\_phone\_number);

END;

*Резултат:*



**Процедура за добавяне на нов автомобил:**

CREATE OR REPLACE PROCEDURE AddNewVehicle(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE, i\_make IN vehicles.vehicle\_make%TYPE, i\_model IN vehicles.vehicle\_model%TYPE)

AS

v\_vehicle\_exists NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_vehicle\_exists FROM vehicles WHERE i\_license\_plate = vehicle\_license\_plate;

IF v\_vehicle\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('A vehicle with this license plate already exists in the database.');

ELSE

INSERT INTO vehicles(vehicle\_license\_plate, vehicle\_make, vehicle\_model) VALUES (i\_license\_plate, i\_make, i\_model);

DBMS\_OUTPUT.PUT\_LINE('Vehicle successfully added to the database.');

END IF;

END AddNewVehicle;

*Код за извикване:*

DECLARE

i\_license\_plate vehicles.vehicle\_license\_plate%TYPE := 'CA1234CB';

i\_make vehicles.vehicle\_make%TYPE := 'Honda';

i\_model vehicles.vehicle\_model%TYPE := 'Accord';

BEGIN

vehicle\_service.AddNewVehicle(i\_license\_plate, i\_make, i\_model);

END;

*Резултат:*



**Процедура за добавяне на нов клиент:**

CREATE OR REPLACE PROCEDURE AddNewCustomer(i\_name IN customers.customer\_name%TYPE, i\_address IN customers.customer\_address%TYPE, i\_phone\_number IN customers.customer\_phone\_number%TYPE, i\_vehicle\_license\_plate IN customers.vehicle\_license\_plate%TYPE)

AS

v\_customer\_exists NUMBER;

v\_phone\_number\_taken\_by\_technician NUMBER;

v\_license\_plate\_exists NUMBER;

CHECK\_CONSTRAINT\_VIOLATED EXCEPTION;

PRAGMA EXCEPTION\_INIT(CHECK\_CONSTRAINT\_VIOLATED, -2291);

BEGIN

SELECT COUNT(\*) INTO v\_customer\_exists FROM customers WHERE i\_phone\_number = customer\_phone\_number;

SELECT COUNT(\*) INTO v\_phone\_number\_taken\_by\_technician FROM technicians WHERE i\_phone\_number = technician\_phone\_number;

IF v\_customer\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This phone number is already used by an existing customer.');

ELSIF v\_phone\_number\_taken\_by\_technician > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This phone number is already used by an existing technician.');

ELSE

SELECT COUNT(\*) INTO v\_license\_plate\_exists FROM customers WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

IF v\_license\_plate\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This license plate is already used by another customer''s vehicle.');

ELSE

INSERT INTO customers(customer\_name, customer\_address, customer\_phone\_number, vehicle\_license\_plate) VALUES (i\_name, i\_address, i\_phone\_number, i\_vehicle\_license\_plate);

DBMS\_OUTPUT.PUT\_LINE('Customer successfully added to the database.');

END IF;

END IF;

EXCEPTION

WHEN CHECK\_CONSTRAINT\_VIOLATED THEN

DBMS\_OUTPUT.PUT\_LINE('No vehicle with the given license plate exists in the database.');

END AddNewCustomer;

*Код за извикване:*

DECLARE

i\_name customers.customer\_name%TYPE := 'Tony Stark';

i\_address customers.customer\_address%TYPE := 'Sofia, Mladost 3, bl. 345, entr. 4, fl. 3, ap. 300';

i\_phone\_number customers.customer\_phone\_number%TYPE := '0987654321';

i\_vehicle\_license\_plate customers.vehicle\_license\_plate%TYPE := 'CA1234CB';

BEGIN

vehicle\_service.AddNewCustomer(i\_name, i\_address, i\_phone\_number, i\_vehicle\_license\_plate);

END;

*Резултат:*



**Процедура за добавяне на нов уречен час:**

CREATE OR REPLACE PROCEDURE AddNewAppointment(i\_phone\_number IN appointments.technician\_phone\_number%TYPE, i\_vehicle\_license\_plate IN appointments.vehicle\_license\_plate%TYPE, i\_timestamp IN appointments.appointment\_timestamp%TYPE)

AS

v\_appointment\_exists NUMBER;

v\_technician\_exists NUMBER;

v\_vehicle\_exists NUMBER;

v\_vehicle\_has\_owner NUMBER;

NOT\_TECHNICIAN\_EXISTS EXCEPTION;

NOT\_VEHICLE\_EXISTS EXCEPTION;

NOT\_VEHICLE\_OWNER EXCEPTION;

BEGIN

SELECT COUNT(\*) INTO v\_appointment\_exists FROM appointments WHERE (i\_phone\_number = technician\_phone\_number AND i\_vehicle\_license\_plate = vehicle\_license\_plate) OR i\_timestamp = appointment\_timestamp;

SELECT COUNT(\*) INTO v\_technician\_exists FROM technicians WHERE i\_phone\_number = technician\_phone\_number;

SELECT COUNT(\*) INTO v\_vehicle\_exists FROM vehicles WHERE i\_vehicle\_license\_plate = vehicle\_license\_plate;

SELECT COUNT(\*) INTO v\_vehicle\_has\_owner FROM customers WHERE i\_vehicle\_license\_plate = vehicle\_license\_plate;

IF v\_technician\_exists = 0 THEN

RAISE NOT\_TECHNICIAN\_EXISTS;

END IF;

IF v\_vehicle\_exists = 0 THEN

RAISE NOT\_VEHICLE\_EXISTS;

END IF;

IF v\_vehicle\_has\_owner = 0 THEN

RAISE NOT\_VEHICLE\_OWNER;

END IF;

IF v\_appointment\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This appointment already exists in the database.');

ELSE

INSERT INTO appointments(technician\_phone\_number, vehicle\_license\_plate, appointment\_timestamp) VALUES (i\_phone\_number, i\_vehicle\_license\_plate, i\_timestamp);

DBMS\_OUTPUT.PUT\_LINE('Appointment successfully added to the database.');

END IF;

EXCEPTION

WHEN NOT\_TECHNICIAN\_EXISTS THEN

DBMS\_OUTPUT.PUT\_LINE('Technician with this phone number does not exist in the database.');

WHEN NOT\_VEHICLE\_EXISTS THEN

DBMS\_OUTPUT.PUT\_LINE('Vehicle with this license plate does not exist in the database.');

WHEN NOT\_VEHICLE\_OWNER THEN

DBMS\_OUTPUT.PUT\_LINE('This vehicle does not have an owner assigned to it.');

END AddNewAppointment;

*Код за извикване:*

DECLARE

i\_phone\_number appointments.technician\_phone\_number%TYPE := '1234567890';

i\_vehicle\_license\_plate appointments.vehicle\_license\_plate%TYPE := 'CA1234CB';

i\_date appointments.appointment\_timestamp%TYPE := TO\_TIMESTAMP('29.12.2020 8:30:25', 'DD.MM.YYYY HH24:MI:SS');

BEGIN

vehicle\_service.AddNewAppointment(i\_phone\_number, i\_vehicle\_license\_plate, i\_date);

END;

*Резултат:*



**Функция за показване на всички техници в базата данни:**

CREATE OR REPLACE FUNCTION getAllTechnicians RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM technicians;

RETURN cur;

END getAllTechnicians;

*Код за извикване:*

DECLARE

cur SYS\_REFCURSOR;

v\_technician\_name technicians.technician\_name%TYPE;

v\_technician\_field technicians.technician\_field%TYPE;

v\_technician\_phone\_number technicians.technician\_phone\_number%TYPE;

BEGIN

cur := vehicle\_service.getAllTechnicians();

DBMS\_OUTPUT.PUT\_LINE('All technicians in the service:' || chr(10));

LOOP

FETCH cur INTO v\_technician\_name, v\_technician\_field, v\_technician\_phone\_number;

EXIT WHEN cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Technician name: ' || v\_technician\_name);

DBMS\_OUTPUT.PUT\_LINE('Technician field: ' || v\_technician\_field);

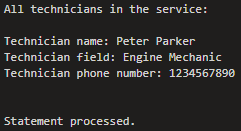
DBMS\_OUTPUT.PUT\_LINE('Technician phone number: ' || v\_technician\_phone\_number || chr(10));

END LOOP;

CLOSE cur;

END;

*Резултат:*



**Функция за показване на всички автомобили в базата данни:**

CREATE OR REPLACE FUNCTION getAllVehicles RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM vehicles;

RETURN cur;

END getAllVehicles;

*Код за извикване:*

DECLARE

cur SYS\_REFCURSOR;

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE;

v\_vehicle\_make vehicles.vehicle\_make%TYPE;

v\_vehicle\_model vehicles.vehicle\_model%TYPE;

BEGIN

cur := vehicle\_service.getAllVehicles();

DBMS\_OUTPUT.PUT\_LINE('All vehicles in the service:' || chr(10));

LOOP

FETCH cur INTO v\_vehicle\_license\_plate, v\_vehicle\_make, v\_vehicle\_model;

EXIT WHEN cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Vehicle license plate: ' || v\_vehicle\_license\_plate);

DBMS\_OUTPUT.PUT\_LINE('Vehicle make: ' || v\_vehicle\_make);

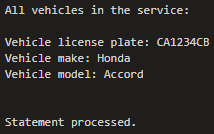
DBMS\_OUTPUT.PUT\_LINE('Vehicle model: ' || v\_vehicle\_model || chr(10));

END LOOP;

CLOSE cur;

END;

*Резултат:*



**Функция за показване на всички клиенти в базата данни:**

CREATE OR REPLACE FUNCTION getAllCustomers RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM customers;

RETURN cur;

END getAllCustomers;

*Код за извикване:*

DECLARE

cur SYS\_REFCURSOR;

v\_customer\_name customers.customer\_name%TYPE;

v\_customer\_address customers.customer\_address%TYPE;

v\_customer\_phone\_number customers.customer\_phone\_number%TYPE;

v\_vehicle\_license\_plate customers.vehicle\_license\_plate%TYPE;

BEGIN

cur := vehicle\_service.getAllCustomers();

DBMS\_OUTPUT.PUT\_LINE('All customers in the service:' || chr(10));

LOOP

FETCH cur INTO v\_customer\_name, v\_customer\_address, v\_customer\_phone\_number, v\_vehicle\_license\_plate;

EXIT WHEN cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Customer name: ' || v\_customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Customer address: ' || v\_customer\_address);

DBMS\_OUTPUT.PUT\_LINE('Customer phone number: ' || v\_customer\_phone\_number);

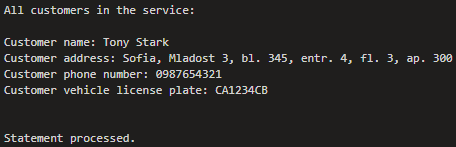
DBMS\_OUTPUT.PUT\_LINE('Customer vehicle license plate: ' || v\_vehicle\_license\_plate || chr(10));

END LOOP;

CLOSE cur;

END;

*Резултат:*



**Функция за показване на името на техник чрез телефонния му номер:**

CREATE OR REPLACE FUNCTION getTechnicianNameFromNumber(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN technicians.technician\_name%TYPE AS tech\_name technicians.technician\_name%TYPE;

BEGIN

SELECT technician\_name INTO tech\_name FROM technicians WHERE technician\_phone\_number = i\_phone\_number;

RETURN tech\_name;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getTechnicianNameFromNumber;

*Код за извикване:*

DECLARE

v\_technician\_phone\_number technicians.technician\_phone\_number%TYPE := 1234567890;

BEGIN

IF vehicle\_service.getTechnicianNameFromNumber(v\_technician\_phone\_number) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Technician with this phone number does not exist in the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Technician name: ' || vehicle\_service.getTechnicianNameFromNumber(v\_technician\_phone\_number));

END IF;

END;

*Резултат:*



**Функция за показване на името на клиент чрез регистрационния номер на автомобила му:**

CREATE OR REPLACE FUNCTION getCustomerNameFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_name%TYPE AS owner\_name customers.customer\_name%TYPE;

BEGIN

SELECT customer\_name INTO owner\_name FROM customers WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN owner\_name;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getCustomerNameFromVehicleLicensePlate;

*Код за извикване:*

DECLARE

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE := 'CA1234CB';

BEGIN

IF vehicle\_service.getCustomerNameFromVehicleLicensePlate(v\_vehicle\_license\_plate) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Vehicle with this license plate does not exist in the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Customer name: ' || vehicle\_service.getCustomerNameFromVehicleLicensePlate(v\_vehicle\_license\_plate));

END IF;

END;

*Резултат:*



**Функция за показване на телефонния номер на клиент чрез регистрационния номер на автомобила му:**

CREATE OR REPLACE FUNCTION getCustomerPhoneNumberFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_phone\_number%TYPE AS owner\_phone\_number customers.customer\_phone\_number%TYPE;

BEGIN

SELECT customer\_phone\_number INTO owner\_phone\_number FROM customers WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN owner\_phone\_number;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getCustomerPhoneNumberFromVehicleLicensePlate;

*Код за извикване:*

DECLARE

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE := 'CA1234CB';

BEGIN

IF vehicle\_service.getCustomerPhoneNumberFromVehicleLicensePlate(v\_vehicle\_license\_plate) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Vehicle with this license plate does not exist in the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Customer phone number: ' || vehicle\_service.getCustomerPhoneNumberFromVehicleLicensePlate(v\_vehicle\_license\_plate));

END IF;

END;

*Резултат:*



**Функция за показване на всички уредени часове:**

CREATE OR REPLACE FUNCTION getAllAppointments RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM appointments;

RETURN cur;

END getAllAppointments;

*Код за извикване:*

DECLARE

cur SYS\_REFCURSOR;

v\_technician\_phone\_number appointments.technician\_phone\_number%TYPE;

v\_vehicle\_license\_plate appointments.vehicle\_license\_plate%TYPE;

v\_appointment\_timestamp appointments.appointment\_timestamp%TYPE;

BEGIN

cur := vehicle\_service.getAllAppointments();

DBMS\_OUTPUT.PUT\_LINE('All appointments in the service:' || chr(10));

LOOP

FETCH cur INTO v\_technician\_phone\_number, v\_vehicle\_license\_plate, v\_appointment\_timestamp;

EXIT WHEN cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Technician name: ' || vehicle\_service.getTechnicianNameFromNumber(v\_technician\_phone\_number));

DBMS\_OUTPUT.PUT\_LINE('Technician phone number: ' || v\_technician\_phone\_number);

DBMS\_OUTPUT.PUT\_LINE('Vehicle license plate: ' || v\_vehicle\_license\_plate);

DBMS\_OUTPUT.PUT\_LINE('Vehicle owner: ' || vehicle\_service.getCustomerNameFromVehicleLicensePlate(v\_vehicle\_license\_plate));

DBMS\_OUTPUT.PUT\_LINE('Vehicle owner phone number: ' || vehicle\_service.getCustomerPhoneNumberFromVehicleLicensePlate(v\_vehicle\_license\_plate));

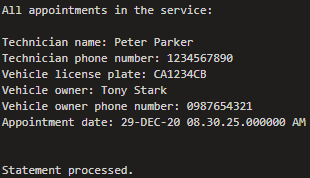
DBMS\_OUTPUT.PUT\_LINE('Appointment date: ' || v\_appointment\_timestamp || chr(10));

END LOOP;

CLOSE cur;

END;

*Резултат:*



**Функция за показване на областта на дейност на техник чрез телефонния му номер:**

CREATE OR REPLACE FUNCTION getTechnicianFieldFromNumber(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN technicians.technician\_field%TYPE AS tech\_field technicians.technician\_field%TYPE;

BEGIN

SELECT technician\_field INTO tech\_field FROM technicians WHERE technician\_phone\_number = i\_phone\_number;

RETURN tech\_field;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getTechnicianFieldFromNumber;

*Код за извикване:*

DECLARE

v\_phone\_number technicians.technician\_phone\_number%TYPE := '1234567890';

BEGIN

IF vehicle\_service.getTechnicianFieldFromNumber(v\_phone\_number) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Technician with this phone number does not exist in the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Technician field: ' || vehicle\_service.getTechnicianFieldFromNumber(v\_phone\_number));

END IF;

END;

*Резултат:*



**Функция за показване на марката на автомобил чрез регистрационния му номер:**

CREATE OR REPLACE FUNCTION getVehicleMakeFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN vehicles.vehicle\_make%TYPE AS veh\_make vehicles.vehicle\_make%TYPE;

BEGIN

SELECT vehicle\_make INTO veh\_make FROM vehicles WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN veh\_make;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getVehicleMakeFromVehicleLicensePlate;

*Код за извикване:*

DECLARE

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE := 'CA1234CB';

BEGIN

IF vehicle\_service.getVehicleMakeFromVehicleLicensePlate(v\_vehicle\_license\_plate) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Vehicle with this license plate does not exist in the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Vehicle make: ' || vehicle\_service.getVehicleMakeFromVehicleLicensePlate(v\_vehicle\_license\_plate));

END IF;

END;

*Резултат:*



**Функция за показване на всички техници, които ще работят по даден автомобил:**

CREATE OR REPLACE FUNCTION getAllTechniciansSignedToVehicle(i\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE) RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM appointments WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

RETURN cur;

END getAllTechniciansSignedToVehicle;

*Код за извикване:*

DECLARE

cur SYS\_REFCURSOR;

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE := 'CA1234CB';

v\_technician\_phone\_number technicians.technician\_phone\_number%TYPE;

v\_appointment\_timestamp appointments.appointment\_timestamp%TYPE;

BEGIN

IF vehicle\_service.getVehicleMakeFromVehicleLicensePlate(v\_vehicle\_license\_plate) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('No vehicle with the given license plate exists in the database.');

RETURN;

END IF;

cur := vehicle\_service.getAllTechniciansSignedToVehicle(v\_vehicle\_license\_plate);

DBMS\_OUTPUT.PUT\_LINE('All technicians signed on to vehicle with license plate ' || v\_vehicle\_license\_plate || ':' || chr(10));

LOOP

FETCH cur INTO v\_technician\_phone\_number, v\_vehicle\_license\_plate, v\_appointment\_timestamp;

EXIT WHEN cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Technician name: ' || vehicle\_service.getTechnicianNameFromNumber(v\_technician\_phone\_number));

DBMS\_OUTPUT.PUT\_LINE('Technician field: ' || vehicle\_service.getTechnicianFieldFromNumber(v\_technician\_phone\_number));

DBMS\_OUTPUT.PUT\_LINE('Technician phone number: ' || v\_technician\_phone\_number);

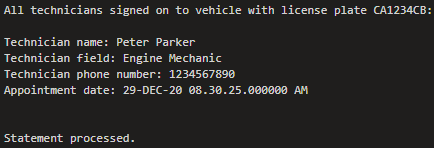
DBMS\_OUTPUT.PUT\_LINE('Appointment date: ' || v\_appointment\_timestamp || chr(10));

END LOOP;

CLOSE cur;

END;

*Резултат:*



**Функция за показване на модела на автомобил чрез регистрационния му номер:**

CREATE OR REPLACE FUNCTION getVehicleModelFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN vehicles.vehicle\_model%TYPE AS veh\_model vehicles.vehicle\_model%TYPE;

BEGIN

SELECT vehicle\_model INTO veh\_model FROM vehicles WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN veh\_model;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getVehicleModelFromVehicleLicensePlate;

*Код за извикване:*

DECLARE

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE := 'CA1234CB';

BEGIN

IF vehicle\_service.getVehicleModelFromVehicleLicensePlate(v\_vehicle\_license\_plate) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Vehicle with this license plate does not exist in the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Vehicle model: ' || vehicle\_service.getVehicleModelFromVehicleLicensePlate(v\_vehicle\_license\_plate));

END IF;

END;

*Резултат:*



**Функция за показване на адреса на клиент чрез регистрационния номер на автомобила му:**

CREATE OR REPLACE FUNCTION getCustomerAddressFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_address%TYPE AS owner\_address customers.customer\_address%TYPE;

BEGIN

SELECT customer\_address INTO owner\_address FROM customers WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN owner\_address;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getCustomerAddressFromVehicleLicensePlate;

*Код за извикване:*

DECLARE

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE := 'CA1234CB';

BEGIN

IF vehicle\_service.getCustomerAddressFromVehicleLicensePlate(v\_vehicle\_license\_plate) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Vehicle with this license plate does not exist in the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Customer address: ' || vehicle\_service.getCustomerAddressFromVehicleLicensePlate(v\_vehicle\_license\_plate));

END IF;

END;

*Резултат:*



**Показване на всички автомобили и техните собственици:**

DECLARE

cur SYS\_REFCURSOR;

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE;

v\_vehicle\_make vehicles.vehicle\_make%TYPE;

v\_vehicle\_model vehicles.vehicle\_model%TYPE;

BEGIN

cur := vehicle\_service.getAllVehicles();

DBMS\_OUTPUT.PUT\_LINE('All vehicles and their owners in the service:' || chr(10));

LOOP

FETCH cur INTO v\_vehicle\_license\_plate, v\_vehicle\_make, v\_vehicle\_model;

EXIT WHEN cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Vehicle license plate: ' || v\_vehicle\_license\_plate);

DBMS\_OUTPUT.PUT\_LINE('Vehicle make: ' || v\_vehicle\_make);

DBMS\_OUTPUT.PUT\_LINE('Vehicle model: ' || v\_vehicle\_model);

DBMS\_OUTPUT.PUT\_LINE('Owner name: ' || vehicle\_service.getCustomerNameFromVehicleLicensePlate(v\_vehicle\_license\_plate));

DBMS\_OUTPUT.PUT\_LINE('Owner phone number: ' || vehicle\_service.getCustomerPhoneNumberFromVehicleLicensePlate(v\_vehicle\_license\_plate));

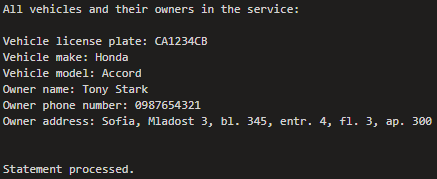
DBMS\_OUTPUT.PUT\_LINE('Owner address: ' || vehicle\_service.getCustomerAddressFromVehicleLicensePlate(v\_vehicle\_license\_plate) || chr(10));

END LOOP;

CLOSE cur;

END;

*Резултат:*



**Функция за показване на всички автомобили и техните собственици, записани при даден техник:**

CREATE OR REPLACE FUNCTION getAllVehiclesWithOwnersSignedToTechnician(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM appointments WHERE technician\_phone\_number = i\_phone\_number;

RETURN cur;

END getAllVehiclesWithOwnersSignedToTechnician;

*Код за извикване:*

DECLARE

cur SYS\_REFCURSOR;

v\_technician\_phone\_number technicians.technician\_phone\_number%TYPE := '1234567890';

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE;

v\_vehicle\_make vehicles.vehicle\_make%TYPE;

v\_vehicle\_model vehicles.vehicle\_model%TYPE;

v\_vehicle\_owner\_name customers.customer\_name%TYPE;

v\_vehicle\_owner\_phone\_number customers.customer\_phone\_number%TYPE;

v\_vehicle\_owner\_address customers.customer\_address%TYPE;

v\_appointment\_timestamp appointments.appointment\_timestamp%TYPE;

BEGIN

IF vehicle\_service.getTechnicianNameFromNumber(v\_technician\_phone\_number) IS NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Technician with this phone number does not exist in the database.');

RETURN;

END IF;

cur := vehicle\_service.getAllVehiclesWithOwnersSignedToTechnician(v\_technician\_phone\_number);

DBMS\_OUTPUT.PUT\_LINE('All vehicles and their owners signed on to technician ' || vehicle\_service.getTechnicianNameFromNumber(v\_technician\_phone\_number) || ' (' || v\_technician\_phone\_number || ') in the service:' || chr(10));

LOOP

FETCH cur INTO v\_technician\_phone\_number, v\_vehicle\_license\_plate, v\_appointment\_timestamp;

EXIT WHEN cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Appointment date: ' || v\_appointment\_timestamp);

DBMS\_OUTPUT.PUT\_LINE('Vehicle license plate: ' || v\_vehicle\_license\_plate);

DBMS\_OUTPUT.PUT\_LINE('Vehicle make: ' || vehicle\_service.getVehicleMakeFromVehicleLicensePlate(v\_vehicle\_license\_plate));

DBMS\_OUTPUT.PUT\_LINE('Vehicle model: ' || vehicle\_service.getVehicleModelFromVehicleLicensePlate(v\_vehicle\_license\_plate));

DBMS\_OUTPUT.PUT\_LINE('Owner name: ' || vehicle\_service.getCustomerNameFromVehicleLicensePlate(v\_vehicle\_license\_plate));

DBMS\_OUTPUT.PUT\_LINE('Owner phone number: ' || vehicle\_service.getCustomerPhoneNumberFromVehicleLicensePlate(v\_vehicle\_license\_plate));

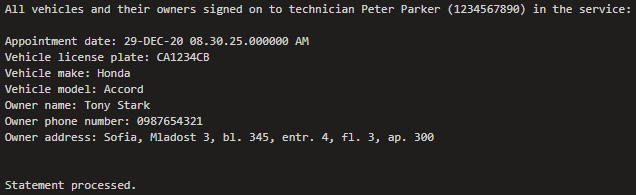
DBMS\_OUTPUT.PUT\_LINE('Owner address: ' || vehicle\_service.getCustomerAddressFromVehicleLicensePlate(v\_vehicle\_license\_plate) || chr(10));

END LOOP;

CLOSE cur;

END;

*Резултат:*



**Функция за проверка дали дадена дата съществува като уреден час в базата данни:**

CREATE OR REPLACE FUNCTION checkTimestampExistsFromDate(i\_date IN VARCHAR2) RETURN NUMBER AS appointment\_exists NUMBER;

BEGIN

SELECT COUNT(\*) INTO appointment\_exists FROM appointments WHERE SUBSTR(appointment\_timestamp, 0, 9) = SUBSTR(TO\_TIMESTAMP(i\_date, 'DD.MM.YYYY'), 0, 9);

RETURN appointment\_exists;

END checkTimestampExistsFromDate;

*Код за извикване:*

DECLARE

v\_date VARCHAR(10) := '29.12.2020';

BEGIN

IF vehicle\_service.checkTimestampExistsFromDate(v\_date) = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Appointment date does not exist in the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Appointments on ' || v\_date || ': ' || vehicle\_service.checkTimestampExistsFromDate(v\_date));

END IF;

END;

*Резултат:*



**Функция за показване на всички автомобили и техните собственици, записани на дадена дата:**

CREATE OR REPLACE FUNCTION getAllVehiclesWithOwnersOnSpecificDate(i\_date IN VARCHAR2) RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM appointments WHERE SUBSTR(appointment\_timestamp, 0, 9) = SUBSTR(TO\_TIMESTAMP(i\_date, 'DD.MM.YYYY'), 0, 9);

RETURN cur;

END getAllVehiclesWithOwnersOnSpecificDate;

*Код за извикване:*

DECLARE

cur SYS\_REFCURSOR;

v\_date VARCHAR2(10) := '29.12.2020';

v\_technician\_phone\_number technicians.technician\_phone\_number%TYPE;

v\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE;

v\_vehicle\_make vehicles.vehicle\_make%TYPE;

v\_vehicle\_model vehicles.vehicle\_model%TYPE;

v\_vehicle\_owner\_name customers.customer\_name%TYPE;

v\_vehicle\_owner\_phone\_number customers.customer\_phone\_number%TYPE;

v\_vehicle\_owner\_address customers.customer\_address%TYPE;

v\_appointment\_timestamp appointments.appointment\_timestamp%TYPE;

BEGIN

IF vehicle\_service.checkTimestampExistsFromDate(v\_date) = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Appointment date does not exist in the database.');

RETURN;

END IF;

cur := vehicle\_service.getAllVehiclesWithOwnersOnSpecificDate(v\_date);

DBMS\_OUTPUT.PUT\_LINE('All vehicles, their owners and the technicians signed on to them on ' || v\_date || ' in the service:' || chr(10));

LOOP

FETCH cur INTO v\_technician\_phone\_number, v\_vehicle\_license\_plate, v\_appointment\_timestamp;

EXIT WHEN cur%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Appointment date: ' || v\_appointment\_timestamp);

DBMS\_OUTPUT.PUT\_LINE('Technician name: ' || vehicle\_service.getTechnicianNameFromNumber(v\_technician\_phone\_number));

DBMS\_OUTPUT.PUT\_LINE('Technician field: ' || vehicle\_service.getTechnicianFieldFromNumber(v\_technician\_phone\_number));

DBMS\_OUTPUT.PUT\_LINE('Technician phone number: ' || v\_technician\_phone\_number);

DBMS\_OUTPUT.PUT\_LINE('Vehicle license plate: ' || v\_vehicle\_license\_plate);

DBMS\_OUTPUT.PUT\_LINE('Vehicle make: ' || vehicle\_service.getVehicleMakeFromVehicleLicensePlate(v\_vehicle\_license\_plate));

DBMS\_OUTPUT.PUT\_LINE('Vehicle model: ' || vehicle\_service.getVehicleModelFromVehicleLicensePlate(v\_vehicle\_license\_plate));

DBMS\_OUTPUT.PUT\_LINE('Owner name: ' || vehicle\_service.getCustomerNameFromVehicleLicensePlate(v\_vehicle\_license\_plate));

DBMS\_OUTPUT.PUT\_LINE('Owner phone number: ' || vehicle\_service.getCustomerPhoneNumberFromVehicleLicensePlate(v\_vehicle\_license\_plate));

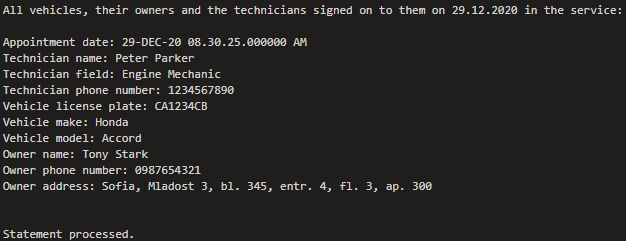
DBMS\_OUTPUT.PUT\_LINE('Owner address: ' || vehicle\_service.getCustomerAddressFromVehicleLicensePlate(v\_vehicle\_license\_plate) || chr(10));

END LOOP;

CLOSE cur;

END;

*Резултат:*



**Процедура за премахване на уреден час от базата данни:**

CREATE OR REPLACE PROCEDURE RemoveExistingAppointment(i\_vehicle\_license\_plate appointments.vehicle\_license\_plate%TYPE)

AS

v\_appointment\_exists NUMBER;

v\_vehicle\_exists NUMBER;

NOT\_VEHICLE\_EXISTS EXCEPTION;

BEGIN

SELECT COUNT(\*) INTO v\_appointment\_exists FROM appointments WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

SELECT COUNT(\*) INTO v\_vehicle\_exists FROM vehicles WHERE i\_vehicle\_license\_plate = vehicle\_license\_plate;

IF v\_vehicle\_exists = 0 THEN

RAISE NOT\_VEHICLE\_EXISTS;

END IF;

IF v\_appointment\_exists > 0 THEN

DELETE FROM appointments WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

DBMS\_OUTPUT.PUT\_LINE('Appointment successfully removed from the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('No such appointment exists in the database.');

END IF;

EXCEPTION

WHEN NOT\_VEHICLE\_EXISTS THEN

DBMS\_OUTPUT.PUT\_LINE('No vehicle with the given license plate exists in the database.');

END RemoveExistingAppointment;

*Код за извикване:*

DECLARE

i\_vehicle\_license\_plate appointments.vehicle\_license\_plate%TYPE := 'CA1234CB';

BEGIN

vehicle\_service.RemoveExistingAppointment(i\_vehicle\_license\_plate);

END;

*Резултат:*



**Процедура за премахване на клиент от базата данни:**

CREATE OR REPLACE PROCEDURE RemoveExistingCustomer(i\_phone\_number IN customers.customer\_phone\_number%TYPE DEFAULT NULL, i\_vehicle\_license\_plate customers.vehicle\_license\_plate%TYPE DEFAULT NULL)

AS

v\_customer\_exists NUMBER;

v\_vehicle\_license\_plate customers.vehicle\_license\_plate%TYPE;

BEGIN

IF (i\_phone\_number IS NULL AND i\_vehicle\_license\_plate IS NULL) OR (i\_phone\_number IS NOT NULL AND i\_vehicle\_license\_plate IS NOT NULL) THEN

DBMS\_OUTPUT.PUT\_LINE('Please input either customer phone number or vehicle license plate.');

RETURN;

END IF;

IF i\_phone\_number IS NOT NULL THEN

SELECT COUNT(\*) INTO v\_customer\_exists FROM customers WHERE customer\_phone\_number = i\_phone\_number;

ELSIF i\_vehicle\_license\_plate IS NOT NULL THEN

SELECT COUNT(\*) INTO v\_customer\_exists FROM customers WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

END IF;

IF v\_customer\_exists > 0 THEN

IF i\_phone\_number IS NOT NULL THEN

SELECT vehicle\_license\_plate INTO v\_vehicle\_license\_plate FROM customers WHERE customer\_phone\_number = i\_phone\_number;

ELSIF i\_vehicle\_license\_plate IS NOT NULL THEN

SELECT vehicle\_license\_plate INTO v\_vehicle\_license\_plate FROM customers WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

END IF;

DELETE FROM customers WHERE vehicle\_license\_plate = v\_vehicle\_license\_plate;

DELETE FROM appointments WHERE vehicle\_license\_plate = v\_vehicle\_license\_plate;

DELETE FROM vehicles WHERE vehicle\_license\_plate = v\_vehicle\_license\_plate;

DBMS\_OUTPUT.PUT\_LINE('Customer successfully removed from the database.');

ELSE

IF i\_phone\_number IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Customer with this phone number does not exist in the database.');

ELSIF i\_vehicle\_license\_plate IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Customer with this license plate does not exist in the database.');

END IF;

END IF;

END RemoveExistingCustomer;

*Код за извикване:*

DECLARE

i\_phone\_number customers.customer\_phone\_number%TYPE := '1234567890';

i\_vehicle\_license\_plate customers.vehicle\_license\_plate%TYPE := 'CB4321CA';

BEGIN

vehicle\_service.RemoveExistingCustomer(i\_phone\_number => i\_phone\_number);

--vehicle\_service.RemoveExistingCustomer(i\_vehicle\_license\_plate => i\_vehicle\_license\_plate);

END;

*Резултат:*



**Процедура за премахване на техник от базата данни:**

CREATE OR REPLACE PROCEDURE RemoveExistingTechnician(i\_phone\_number IN technicians.technician\_phone\_number%TYPE)

AS

v\_technician\_exists NUMBER;

v\_appointments\_exist NUMBER;

v\_extra\_text VARCHAR2(22) := '';

BEGIN

SELECT COUNT(\*) INTO v\_technician\_exists FROM technicians WHERE technician\_phone\_number = i\_phone\_number;

IF v\_technician\_exists > 0 THEN

SELECT COUNT(\*) INTO v\_appointments\_exist FROM appointments WHERE technician\_phone\_number = i\_phone\_number;

IF v\_appointments\_exist > 0 THEN

DELETE FROM appointments WHERE technician\_phone\_number = i\_phone\_number;

v\_extra\_text := 'and their appointments ';

END IF;

DELETE FROM technicians WHERE technician\_phone\_number = i\_phone\_number;

DBMS\_OUTPUT.PUT\_LINE('Technician ' || v\_extra\_text || 'successfully removed from the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Technician with this phone number does not exist in the database.');

END IF;

END RemoveExistingTechnician;

*Код за извикване:*

DECLARE

i\_phone\_number technicians.technician\_phone\_number%TYPE := '1234567890';

BEGIN

vehicle\_service.RemoveExistingTechnician(i\_phone\_number);

END;

*Резултат:*



**Процедура за премахване на автомобил от базата данни:**

CREATE OR REPLACE PROCEDURE RemoveExistingVehicle(i\_vehicle\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE)

AS

v\_vehicle\_exists NUMBER;

v\_customer\_exists NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_vehicle\_exists FROM vehicles WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

IF v\_vehicle\_exists > 0 THEN

SELECT COUNT(\*) INTO v\_customer\_exists FROM customers WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

IF v\_customer\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Cannot remove vehicle as it is linked to an existing customer.');

ELSE

DELETE FROM vehicles WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

DBMS\_OUTPUT.PUT\_LINE('Vehicle successfully removed from the database.');

END IF;

ELSE

DBMS\_OUTPUT.PUT\_LINE('No vehicle with the given license plate exists in the database.');

END IF;

END RemoveExistingVehicle;

*Код за извикване:*

DECLARE

i\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE := 'CA1234CB';

BEGIN

vehicle\_service.RemoveExistingVehicle(i\_vehicle\_license\_plate);

END;

*Резултат:*



**Пакет:**

CREATE OR REPLACE PACKAGE vehicle\_service

AS

PROCEDURE AddNewTechnician(i\_name IN technicians.technician\_name%TYPE, i\_field IN technicians.technician\_field%TYPE, i\_phone\_number IN technicians.technician\_phone\_number%TYPE);

PROCEDURE AddNewVehicle(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE, i\_make IN vehicles.vehicle\_make%TYPE, i\_model IN vehicles.vehicle\_model%TYPE);

PROCEDURE AddNewCustomer(i\_name IN customers.customer\_name%TYPE, i\_address IN customers.customer\_address%TYPE, i\_phone\_number IN customers.customer\_phone\_number%TYPE, i\_vehicle\_license\_plate IN customers.vehicle\_license\_plate%TYPE);

PROCEDURE AddNewAppointment(i\_phone\_number IN appointments.technician\_phone\_number%TYPE, i\_vehicle\_license\_plate IN appointments.vehicle\_license\_plate%TYPE, i\_timestamp IN appointments.appointment\_timestamp%TYPE);

FUNCTION getAllTechnicians RETURN SYS\_REFCURSOR;

FUNCTION getAllVehicles RETURN SYS\_REFCURSOR;

FUNCTION getAllCustomers RETURN SYS\_REFCURSOR;

FUNCTION getTechnicianNameFromNumber(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN technicians.technician\_name%TYPE;

FUNCTION getCustomerNameFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_name%TYPE;

FUNCTION getCustomerPhoneNumberFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_phone\_number%TYPE;

FUNCTION getAllAppointments RETURN SYS\_REFCURSOR;

FUNCTION getTechnicianFieldFromNumber(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN technicians.technician\_field%TYPE;

FUNCTION getVehicleMakeFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN vehicles.vehicle\_make%TYPE;

FUNCTION getAllTechniciansSignedToVehicle(i\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE) RETURN SYS\_REFCURSOR;

FUNCTION getVehicleModelFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN vehicles.vehicle\_model%TYPE;

FUNCTION getCustomerAddressFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_address%TYPE;

FUNCTION getAllVehiclesWithOwnersSignedToTechnician(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN SYS\_REFCURSOR;

FUNCTION checkTimestampExistsFromDate(i\_date IN VARCHAR2) RETURN NUMBER;

FUNCTION getAllVehiclesWithOwnersOnSpecificDate(i\_date IN VARCHAR2) RETURN SYS\_REFCURSOR;

PROCEDURE RemoveExistingAppointment(i\_vehicle\_license\_plate appointments.vehicle\_license\_plate%TYPE);

PROCEDURE RemoveExistingCustomer(i\_phone\_number IN customers.customer\_phone\_number%TYPE DEFAULT NULL, i\_vehicle\_license\_plate customers.vehicle\_license\_plate%TYPE DEFAULT NULL);

PROCEDURE RemoveExistingTechnician(i\_phone\_number IN technicians.technician\_phone\_number%TYPE);

PROCEDURE RemoveExistingVehicle(i\_vehicle\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE);

END vehicle\_service;

/

CREATE OR REPLACE PACKAGE BODY vehicle\_service

AS

--AddNewTechnician

PROCEDURE AddNewTechnician(i\_name IN technicians.technician\_name%TYPE, i\_field IN technicians.technician\_field%TYPE, i\_phone\_number IN technicians.technician\_phone\_number%TYPE)

AS

v\_technician\_exists NUMBER;

v\_phone\_number\_taken\_by\_customer NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_technician\_exists FROM technicians WHERE i\_phone\_number = technician\_phone\_number;

SELECT COUNT(\*) INTO v\_phone\_number\_taken\_by\_customer FROM customers WHERE i\_phone\_number = customer\_phone\_number;

IF v\_technician\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This phone number is already used by an existing technician.');

ELSIF v\_phone\_number\_taken\_by\_customer > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This phone number is already used by an existing customer.');

ELSE

INSERT INTO technicians(technician\_name, technician\_field, technician\_phone\_number) VALUES (i\_name, i\_field, i\_phone\_number);

DBMS\_OUTPUT.PUT\_LINE('Technician successfully added to the database.');

END IF;

END AddNewTechnician;

--AddNewVehicle

PROCEDURE AddNewVehicle(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE, i\_make IN vehicles.vehicle\_make%TYPE, i\_model IN vehicles.vehicle\_model%TYPE)

AS

v\_vehicle\_exists NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_vehicle\_exists FROM vehicles WHERE i\_license\_plate = vehicle\_license\_plate;

IF v\_vehicle\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('A vehicle with this license plate already exists in the database.');

ELSE

INSERT INTO vehicles(vehicle\_license\_plate, vehicle\_make, vehicle\_model) VALUES (i\_license\_plate, i\_make, i\_model);

DBMS\_OUTPUT.PUT\_LINE('Vehicle successfully added to the database.');

END IF;

END AddNewVehicle;

--AddNewCustomer

PROCEDURE AddNewCustomer(i\_name IN customers.customer\_name%TYPE, i\_address IN customers.customer\_address%TYPE, i\_phone\_number IN customers.customer\_phone\_number%TYPE, i\_vehicle\_license\_plate IN customers.vehicle\_license\_plate%TYPE)

AS

v\_customer\_exists NUMBER;

v\_phone\_number\_taken\_by\_technician NUMBER;

v\_license\_plate\_exists NUMBER;

CHECK\_CONSTRAINT\_VIOLATED EXCEPTION;

PRAGMA EXCEPTION\_INIT(CHECK\_CONSTRAINT\_VIOLATED, -2291);

BEGIN

SELECT COUNT(\*) INTO v\_customer\_exists FROM customers WHERE i\_phone\_number = customer\_phone\_number;

SELECT COUNT(\*) INTO v\_phone\_number\_taken\_by\_technician FROM technicians WHERE i\_phone\_number = technician\_phone\_number;

IF v\_customer\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This phone number is already used by an existing customer.');

ELSIF v\_phone\_number\_taken\_by\_technician > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This phone number is already used by an existing technician.');

ELSE

SELECT COUNT(\*) INTO v\_license\_plate\_exists FROM customers WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

IF v\_license\_plate\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This license plate is already used by another customer''s vehicle.');

ELSE

INSERT INTO customers(customer\_name, customer\_address, customer\_phone\_number, vehicle\_license\_plate) VALUES (i\_name, i\_address, i\_phone\_number, i\_vehicle\_license\_plate);

DBMS\_OUTPUT.PUT\_LINE('Customer successfully added to the database.');

END IF;

END IF;

EXCEPTION

WHEN CHECK\_CONSTRAINT\_VIOLATED THEN

DBMS\_OUTPUT.PUT\_LINE('No vehicle with the given license plate exists in the database.');

END AddNewCustomer;

--AddNewAppointment

PROCEDURE AddNewAppointment(i\_phone\_number IN appointments.technician\_phone\_number%TYPE, i\_vehicle\_license\_plate IN appointments.vehicle\_license\_plate%TYPE, i\_timestamp IN appointments.appointment\_timestamp%TYPE)

AS

v\_appointment\_exists NUMBER;

v\_technician\_exists NUMBER;

v\_vehicle\_exists NUMBER;

v\_vehicle\_has\_owner NUMBER;

NOT\_TECHNICIAN\_EXISTS EXCEPTION;

NOT\_VEHICLE\_EXISTS EXCEPTION;

NOT\_VEHICLE\_OWNER EXCEPTION;

BEGIN

SELECT COUNT(\*) INTO v\_appointment\_exists FROM appointments WHERE (i\_phone\_number = technician\_phone\_number AND i\_vehicle\_license\_plate = vehicle\_license\_plate) OR i\_timestamp = appointment\_timestamp;

SELECT COUNT(\*) INTO v\_technician\_exists FROM technicians WHERE i\_phone\_number = technician\_phone\_number;

SELECT COUNT(\*) INTO v\_vehicle\_exists FROM vehicles WHERE i\_vehicle\_license\_plate = vehicle\_license\_plate;

SELECT COUNT(\*) INTO v\_vehicle\_has\_owner FROM customers WHERE i\_vehicle\_license\_plate = vehicle\_license\_plate;

IF v\_technician\_exists = 0 THEN

RAISE NOT\_TECHNICIAN\_EXISTS;

END IF;

IF v\_vehicle\_exists = 0 THEN

RAISE NOT\_VEHICLE\_EXISTS;

END IF;

IF v\_vehicle\_has\_owner = 0 THEN

RAISE NOT\_VEHICLE\_OWNER;

END IF;

IF v\_appointment\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('This appointment already exists in the database.');

ELSE

INSERT INTO appointments(technician\_phone\_number, vehicle\_license\_plate, appointment\_timestamp) VALUES (i\_phone\_number, i\_vehicle\_license\_plate, i\_timestamp);

DBMS\_OUTPUT.PUT\_LINE('Appointment successfully added to the database.');

END IF;

EXCEPTION

WHEN NOT\_TECHNICIAN\_EXISTS THEN

DBMS\_OUTPUT.PUT\_LINE('Technician with this phone number does not exist in the database.');

WHEN NOT\_VEHICLE\_EXISTS THEN

DBMS\_OUTPUT.PUT\_LINE('Vehicle with this license plate does not exist in the database.');

WHEN NOT\_VEHICLE\_OWNER THEN

DBMS\_OUTPUT.PUT\_LINE('This vehicle does not have an owner assigned to it.');

END AddNewAppointment;

--getAllTechnicians

FUNCTION getAllTechnicians RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM technicians;

RETURN cur;

END getAllTechnicians;

--getAllVehicles

FUNCTION getAllVehicles RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM vehicles;

RETURN cur;

END getAllVehicles;

--getAllCustomers

FUNCTION getAllCustomers RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM customers;

RETURN cur;

END getAllCustomers;

--getTechnicianNameFromNumber

FUNCTION getTechnicianNameFromNumber(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN technicians.technician\_name%TYPE AS tech\_name technicians.technician\_name%TYPE;

BEGIN

SELECT technician\_name INTO tech\_name FROM technicians WHERE technician\_phone\_number = i\_phone\_number;

RETURN tech\_name;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getTechnicianNameFromNumber;

--getCustomerNameFromVehicleLicensePlate

FUNCTION getCustomerNameFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_name%TYPE AS owner\_name customers.customer\_name%TYPE;

BEGIN

SELECT customer\_name INTO owner\_name FROM customers WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN owner\_name;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getCustomerNameFromVehicleLicensePlate;

--getCustomerPhoneNumberFromVehicleLicensePlate

FUNCTION getCustomerPhoneNumberFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_phone\_number%TYPE AS owner\_phone\_number customers.customer\_phone\_number%TYPE;

BEGIN

SELECT customer\_phone\_number INTO owner\_phone\_number FROM customers WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN owner\_phone\_number;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getCustomerPhoneNumberFromVehicleLicensePlate;

--getAllAppointments

FUNCTION getAllAppointments RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM appointments;

RETURN cur;

END getAllAppointments;

--getTechnicianFieldFromNumber

FUNCTION getTechnicianFieldFromNumber(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN technicians.technician\_field%TYPE AS tech\_field technicians.technician\_field%TYPE;

BEGIN

SELECT technician\_field INTO tech\_field FROM technicians WHERE technician\_phone\_number = i\_phone\_number;

RETURN tech\_field;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getTechnicianFieldFromNumber;

--getVehicleMakeFromVehicleLicensePlate

FUNCTION getVehicleMakeFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN vehicles.vehicle\_make%TYPE AS veh\_make vehicles.vehicle\_make%TYPE;

BEGIN

SELECT vehicle\_make INTO veh\_make FROM vehicles WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN veh\_make;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getVehicleMakeFromVehicleLicensePlate;

--getAllTechniciansSignedToVehicle

FUNCTION getAllTechniciansSignedToVehicle(i\_vehicle\_license\_plate vehicles.vehicle\_license\_plate%TYPE) RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM appointments WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

RETURN cur;

END getAllTechniciansSignedToVehicle;

--getVehicleModelFromVehicleLicensePlate

FUNCTION getVehicleModelFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN vehicles.vehicle\_model%TYPE AS veh\_model vehicles.vehicle\_model%TYPE;

BEGIN

SELECT vehicle\_model INTO veh\_model FROM vehicles WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN veh\_model;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getVehicleModelFromVehicleLicensePlate;

--getCustomerAddressFromVehicleLicensePlate

FUNCTION getCustomerAddressFromVehicleLicensePlate(i\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE) RETURN customers.customer\_address%TYPE AS owner\_address customers.customer\_address%TYPE;

BEGIN

SELECT customer\_address INTO owner\_address FROM customers WHERE vehicle\_license\_plate = i\_license\_plate;

RETURN owner\_address;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN '';

END getCustomerAddressFromVehicleLicensePlate;

--getAllVehiclesWithOwnersSignedToTechnician

FUNCTION getAllVehiclesWithOwnersSignedToTechnician(i\_phone\_number IN technicians.technician\_phone\_number%TYPE) RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM appointments WHERE technician\_phone\_number = i\_phone\_number;

RETURN cur;

END getAllVehiclesWithOwnersSignedToTechnician;

--checkTimestampExistsFromDate

FUNCTION checkTimestampExistsFromDate(i\_date IN VARCHAR2) RETURN NUMBER AS appointment\_exists NUMBER;

BEGIN

SELECT COUNT(\*) INTO appointment\_exists FROM appointments WHERE SUBSTR(appointment\_timestamp, 0, 9) = SUBSTR(TO\_TIMESTAMP(i\_date, 'DD.MM.YYYY'), 0, 9);

RETURN appointment\_exists;

END checkTimestampExistsFromDate;

--getAllVehiclesWithOwnersOnSpecificDate

FUNCTION getAllVehiclesWithOwnersOnSpecificDate(i\_date IN VARCHAR2) RETURN SYS\_REFCURSOR AS cur SYS\_REFCURSOR;

BEGIN

OPEN cur FOR SELECT \* FROM appointments WHERE SUBSTR(appointment\_timestamp, 0, 9) = SUBSTR(TO\_TIMESTAMP(i\_date, 'DD.MM.YYYY'), 0, 9);

RETURN cur;

END getAllVehiclesWithOwnersOnSpecificDate;

--RemoveExistingAppointment

PROCEDURE RemoveExistingAppointment(i\_vehicle\_license\_plate appointments.vehicle\_license\_plate%TYPE)

AS

v\_appointment\_exists NUMBER;

v\_vehicle\_exists NUMBER;

NOT\_VEHICLE\_EXISTS EXCEPTION;

BEGIN

SELECT COUNT(\*) INTO v\_appointment\_exists FROM appointments WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

SELECT COUNT(\*) INTO v\_vehicle\_exists FROM vehicles WHERE i\_vehicle\_license\_plate = vehicle\_license\_plate;

IF v\_vehicle\_exists = 0 THEN

RAISE NOT\_VEHICLE\_EXISTS;

END IF;

IF v\_appointment\_exists > 0 THEN

DELETE FROM appointments WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

DBMS\_OUTPUT.PUT\_LINE('Appointment successfully removed from the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('No such appointment exists in the database.');

END IF;

EXCEPTION

WHEN NOT\_VEHICLE\_EXISTS THEN

DBMS\_OUTPUT.PUT\_LINE('No vehicle with the given license plate exists in the database.');

END RemoveExistingAppointment;

--RemoveExistingCustomer

PROCEDURE RemoveExistingCustomer(i\_phone\_number IN customers.customer\_phone\_number%TYPE DEFAULT NULL, i\_vehicle\_license\_plate customers.vehicle\_license\_plate%TYPE DEFAULT NULL)

AS

v\_customer\_exists NUMBER;

v\_vehicle\_license\_plate customers.vehicle\_license\_plate%TYPE;

BEGIN

IF (i\_phone\_number IS NULL AND i\_vehicle\_license\_plate IS NULL) OR (i\_phone\_number IS NOT NULL AND i\_vehicle\_license\_plate IS NOT NULL) THEN

DBMS\_OUTPUT.PUT\_LINE('Please input either customer phone number or vehicle license plate.');

RETURN;

END IF;

IF i\_phone\_number IS NOT NULL THEN

SELECT COUNT(\*) INTO v\_customer\_exists FROM customers WHERE customer\_phone\_number = i\_phone\_number;

ELSIF i\_vehicle\_license\_plate IS NOT NULL THEN

SELECT COUNT(\*) INTO v\_customer\_exists FROM customers WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

END IF;

IF v\_customer\_exists > 0 THEN

IF i\_phone\_number IS NOT NULL THEN

SELECT vehicle\_license\_plate INTO v\_vehicle\_license\_plate FROM customers WHERE customer\_phone\_number = i\_phone\_number;

ELSIF i\_vehicle\_license\_plate IS NOT NULL THEN

SELECT vehicle\_license\_plate INTO v\_vehicle\_license\_plate FROM customers WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

END IF;

DELETE FROM customers WHERE vehicle\_license\_plate = v\_vehicle\_license\_plate;

DELETE FROM appointments WHERE vehicle\_license\_plate = v\_vehicle\_license\_plate;

DELETE FROM vehicles WHERE vehicle\_license\_plate = v\_vehicle\_license\_plate;

DBMS\_OUTPUT.PUT\_LINE('Customer successfully removed from the database.');

ELSE

IF i\_phone\_number IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Customer with this phone number does not exist in the database.');

ELSIF i\_vehicle\_license\_plate IS NOT NULL THEN

DBMS\_OUTPUT.PUT\_LINE('Customer with this license plate does not exist in the database.');

END IF;

END IF;

END RemoveExistingCustomer;

--RemoveExistingTechnician

PROCEDURE RemoveExistingTechnician(i\_phone\_number IN technicians.technician\_phone\_number%TYPE)

AS

v\_technician\_exists NUMBER;

v\_appointments\_exist NUMBER;

v\_extra\_text VARCHAR2(22) := '';

BEGIN

SELECT COUNT(\*) INTO v\_technician\_exists FROM technicians WHERE technician\_phone\_number = i\_phone\_number;

IF v\_technician\_exists > 0 THEN

SELECT COUNT(\*) INTO v\_appointments\_exist FROM appointments WHERE technician\_phone\_number = i\_phone\_number;

IF v\_appointments\_exist > 0 THEN

DELETE FROM appointments WHERE technician\_phone\_number = i\_phone\_number;

v\_extra\_text := 'and their appointments ';

END IF;

DELETE FROM technicians WHERE technician\_phone\_number = i\_phone\_number;

DBMS\_OUTPUT.PUT\_LINE('Technician ' || v\_extra\_text || 'successfully removed from the database.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Technician with this phone number does not exist in the database.');

END IF;

END RemoveExistingTechnician;

--RemoveExistingVehicle

PROCEDURE RemoveExistingVehicle(i\_vehicle\_license\_plate IN vehicles.vehicle\_license\_plate%TYPE)

AS

v\_vehicle\_exists NUMBER;

v\_customer\_exists NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_vehicle\_exists FROM vehicles WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

IF v\_vehicle\_exists > 0 THEN

SELECT COUNT(\*) INTO v\_customer\_exists FROM customers WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

IF v\_customer\_exists > 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Cannot remove vehicle as it is linked to an existing customer.');

ELSE

DELETE FROM vehicles WHERE vehicle\_license\_plate = i\_vehicle\_license\_plate;

DBMS\_OUTPUT.PUT\_LINE('Vehicle successfully removed from the database.');

END IF;

ELSE

DBMS\_OUTPUT.PUT\_LINE('No vehicle with the given license plate exists in the database.');

END IF;

END RemoveExistingVehicle;

END vehicle\_service;

**Workspace: F83533**

**Username: gtuparov**

**Password: gtuparov2021**