JEGYZŐKÖNYV

Modern adatbázis rendszerek Msc

2022. tavasz féléves feladat

Oracle ORDBMS

Készítette: **Polonkai Dávid**
Neptunkód: **GPNWZT**

Modern Adatbázis rendszerek hallgatói jegyzőkönyv

1. Car objektum
    a. Hozz létre egy car objektumot (registration_number, color, production_year, manufacturer, price) attribútumokkal.
    b. Hozz létre egy táblát, amelynek neve car_showroom (car_id, car)
    c. Adj hozzá adatokat
    d. Kérdezd le minden car objektum színét a car_showroom táblából
    e. Állítsd be, hogy minden car fehér legyen ami 5 évnél idősebb
    f. (szorgalmi) * Oldjuk meg ezt member procedure-val *

2. Nested table
    a. Hozd létre a car_table_type tábla típust, ami car-okat tartalmaz
    b. Hozd létre a showroom_network táblát, ami (id, city, car_table) adattagokat tartalmaz
    c. Vigyél fel egy ‘opel’-t és ‘skoda’-t a Miskolc városához tartozó táblába.
    d. Vigyél fel egy ‘opel’-t a Debrecen városához tartozó táblába.
    e. Kérdezzük le a Miskolci szalonban található autókat
    f. Kérdezzük le minden adatot, figyeljük meg a kimenetét
    g. Kérdezzük le a nested tábla adatait és figyeljük meg mit találunk

3. OOP
    a. Hozzuk létre a vehicle_o osztályt
    Adattagok:
        - kerekek száma
        - teljesítmény

    Függvények:
        - power_by_wheel(loss_percent)
        - get_power()

    Eljárások:
        - write_out_wheel_number()
        - write_out_power()
        - set_power(int kw)

    b. Hozzunk létre egy car_o objektumot a vehicle_o osztály gyerekeként,
    Adattagok:
        - registration_number
        - color
        - production_year
        - manufacturer
        - price

    Statikus eljárás:
        - write_object_name_sring()

    Függvény:
        - get_age()

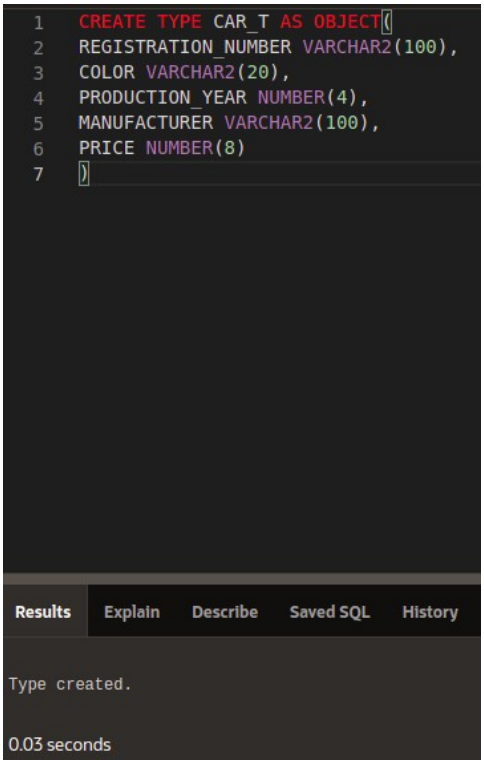
Legyen konstruktor!

c. Próbáljuk ki egy névtelen plsql blokkon belül a létrehozott típus minden függvényét és eljárását.

Modern Adatbázis rendszerek gyakorlatvezetői jegyzőkönyv

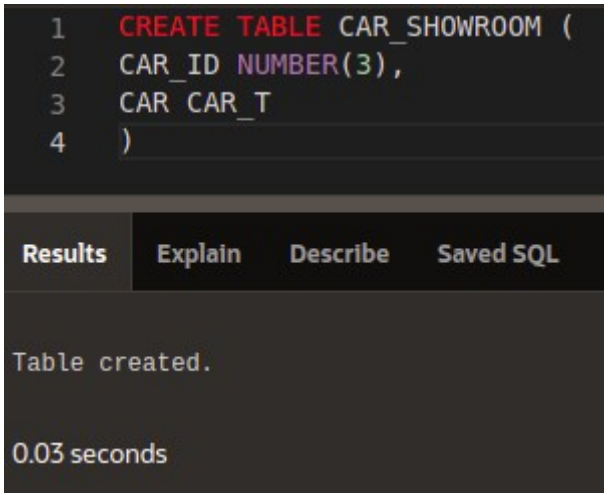
1. Car objektum
    a. Hozz létre egy car objektumot (registration_number, color, production_year, manufacturer, price) attribútumokkal.

```
CREATE TYPE CAR_T AS OBJECT(
REGISTRATION_NUMBER VARCHAR2(100),
COLOR VARCHAR2(20),
PRODUCTION_YEAR NUMBER(4),
MANUFACTURER VARCHAR2(100),
PRICE NUMBER(8)
)
```



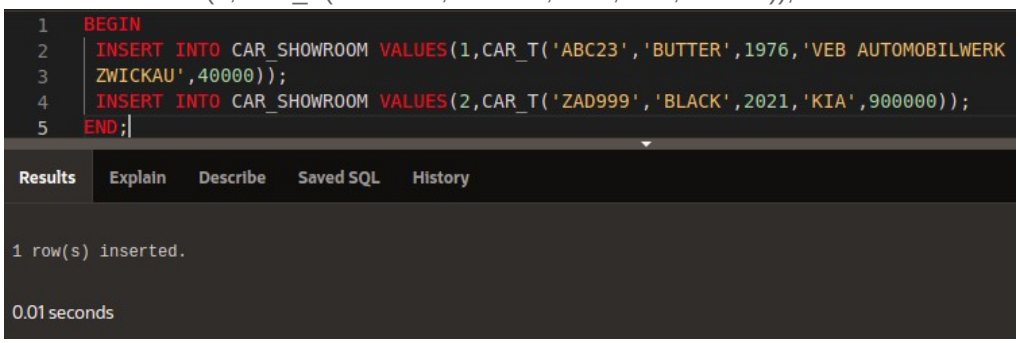
b. Hozz létre egy táblát, amelynek neve car_showroom (car_id, car)

```
CREATE TABLE CAR_SHOWROOM (
CAR_ID NUMBER(3),
CAR CAR_T
)
```



c. Adj hozzá adatokat

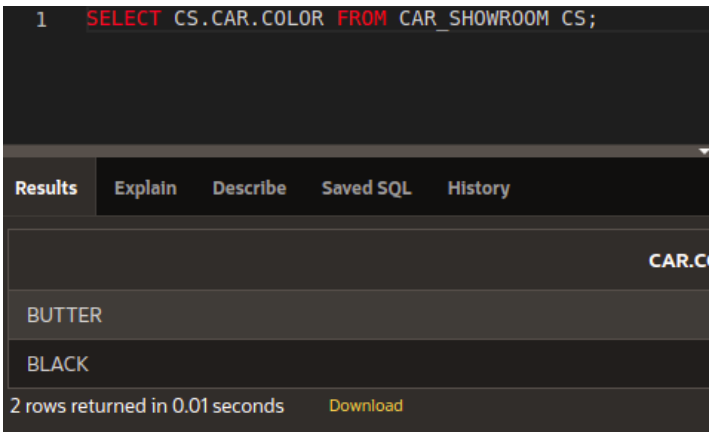
```
INSERT INTO CAR_SHOWROOM VALUES(1,CAR_T('ABC23','BUTTER',1976,'VEB
AUTOMOBILWERK
ZWICKAU',40000));
INSERT INTO CAR_SHOWROOM
VALUES(2,CAR_T('ZAD999','BLACK',2021,'KIA',900000));
```



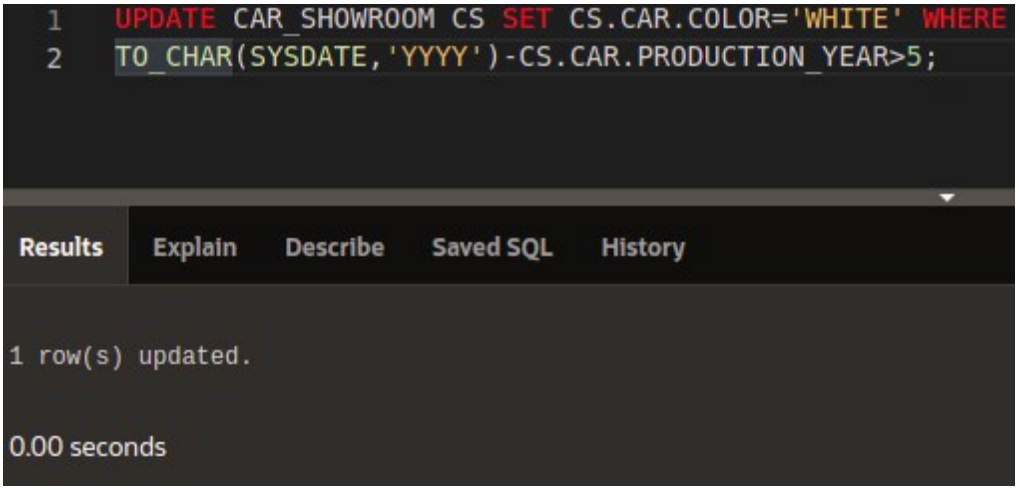
(A nevesítetlen plsql blokk miatt csak az utolsó művelet kimenetét írja ki a consolera az oracle szerver)

d. Kérdezd le minden car objektum színét a car_showroom táblából

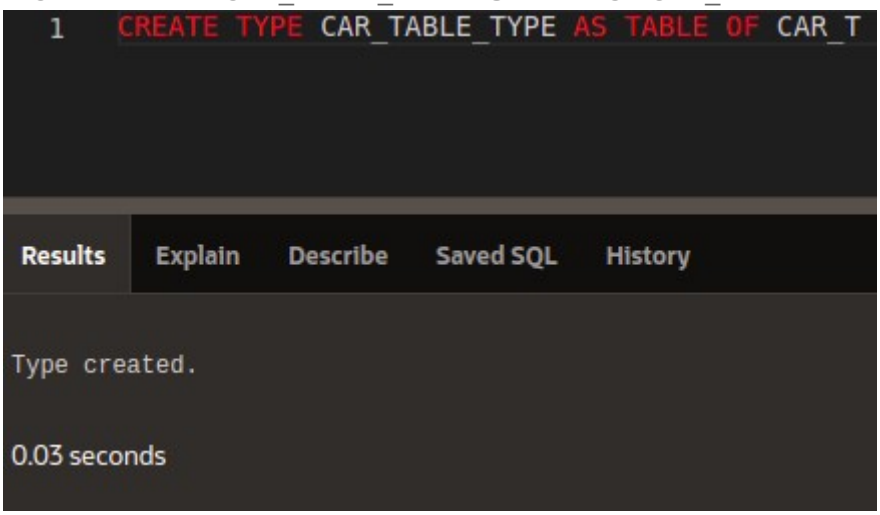
```
SELECT CS.CAR.COLOR FROM CAR_SHOWROOM CS;
```



e. Állítsd be, hogy minden car fehér legyen ami 5 évnél idősebb
UPDATE CAR_SHOWROOM CS SET CS.CAR.COLOR='WHITE' WHERE TO_CHAR(SYSDATE,'YYYY')-CS.CAR.PRODUCTION_YEAR>5;



2. Nested table
   a. Hozd létre a car_table_type tábla típust, ami car-okat tartalmaz
   CREATE TYPE CAR_TABLE_TYPE AS TABLE OF CAR_T



   b. Hozd létre a showroom_network táblát, ami (id, city, car_table) adattagokat tartalmaz

```
CREATE TABLE SHOWROOM_NETWORK(
   ID INT PRIMARY KEY,
   CITY VARCHAR(100),
   CAR_TABLE CAR_TABLE_TYPE
)NESTED TABLE CAR_TABLE STORE AS CAR_NESTED
```

```
CREATE TABLE SHOWROOM_NETWORK(
    ID INT PRIMARY KEY,
    CITY VARCHAR(100),
    CAR_TABLE CAR_TABLE_TYPE
)NESTED TABLE CAR_TABLE STORE AS CAR_NESTED
```

Results | Explain | Describe | Saved SQL | History

```
Table created.

0.06 seconds
```

c. Vigyél fel egy ‘opel’-t és ‘skoda’-t a Miskolc városához tartozó táblába.

```
INSERT INTO SHOWROOM_NETWORK
VALUES(1,'MISKOLC',
CAR_TABLE_TYPE(
   CAR_T('AAA100','GREEN',2000,'OPEL',200000),
   CAR_T('AAA101','BLUE',2010,'SKODA',200000)))
```

```
INSERT INTO SHOWROOM_NETWORK
VALUES(1,'MISKOLC',
CAR_TABLE_TYPE(
    CAR_T('AAA100','GREEN',2000,'OPEL',200000),
    CAR_T('AAA101','BLUE',2010,'SKODA',200000)))
```

Results | Explain | Describe | Saved SQL | History

```
1 row(s) inserted.

0.07 seconds
```

d. Vigyél fel egy ‘opel’-t a Debrecen városához tartozó táblába.

```
INSERT INTO SHOWROOM_NETWORK
VALUES(2,'DEBRECEN',
CAR_TABLE_TYPE(
   CAR_T('AAA200','GREEN',2000,'OPEL',400000)))
```

```
INSERT INTO SHOWROOM_NETWORK
VALUES(2,'DEBRECEN',
CAR_TABLE_TYPE(
    CAR_T('AAA200','GREEN',2000,'OPEL',400000))
)
```

Results | Explain | Describe | Saved SQL | History

1 row(s) inserted.

0.00 seconds

e. Kérdezzük le a Miskolci szalonban található autókat

SELECT * FROM TABLE(SELECT SN.CAR_TABLE FROM SHOWROOM_NETWORK SN WHERE SN.CITY='MISKOLC')

```
SELECT * FROM TABLE(SELECT SN.CAR_TABLE FROM SHOWROOM_NETWORK SN WHERE SN.CITY='MISKOLC')
```

Results | Explain | Describe | Saved SQL | History

REGISTRATION_NUMBER	COLOR	PRODUCTION_YEAR	MANUFACTURER	PRICE
AAA100	GREEN	2000	OPEL	200000
AAA101	BLUE	2010	SKODA	200000

f. Kérdezzük le minden adatot, figyeljük meg a kimenetét

SELECT * FROM SHOWROOM_NETWORK

```
SELECT * FROM SHOWROOM_NETWORK
```

Results | Explain | Describe | Saved SQL | History

ID	CITY	CAR_TABLE
2	DEBRECEN	[unsupported data type]
1	MISKOLC	[unsupported data type]

A NESTED TÁBLA TÍPUS NEM LÁTHATÓ, CSAK UNSUPPORTED DATATYPEKÉNT, MERT NEM LEHET ILYEN MÓDON LEKÉRDEZNI

g. Kérdezzük le a nested tábla adatait és figyeljük meg mit találunk

SELECT * FROM CAR_NESTED

```
SELECT * FROM CAR_NESTED
```

Results | Explain | Describe | Saved SQL | History

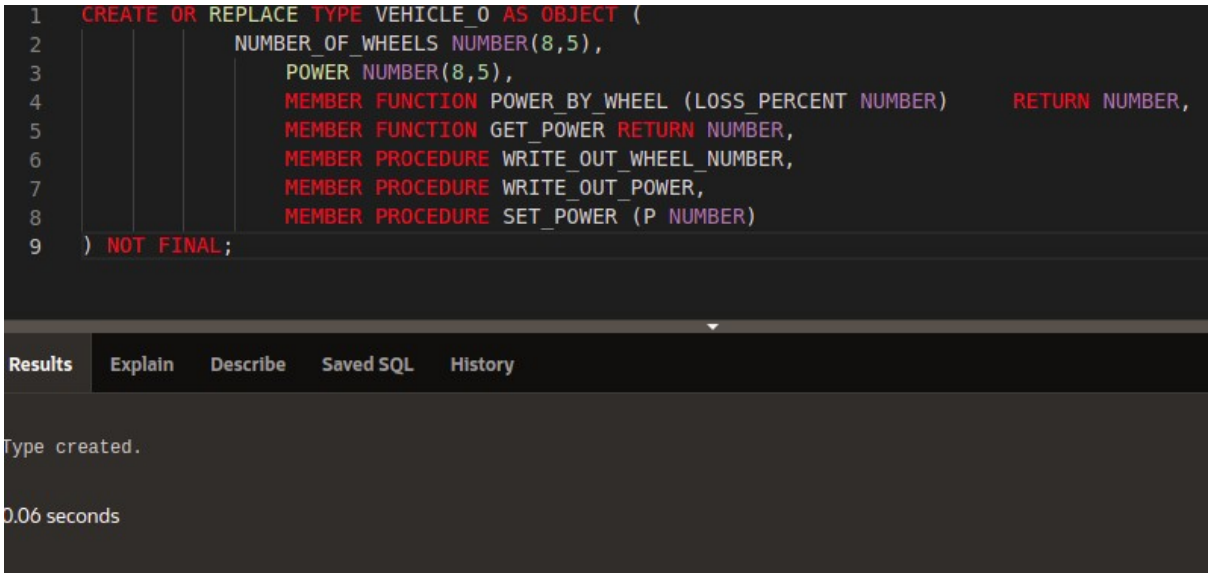
REGISTRATION_NUMBER	COLOR	PRODUCTION_YEAR	MANUFACTURER	PRICE
AAA200	GREEN	2000	OPEL	400000
AAA100	GREEN	2000	OPEL	200000
AAA101	BLUE	2010	SKODA	200000

3. OOP
    a. Hozzuk létre a vehicle_o osztályt
    Adattagok:
        - kerekek száma
        - teljesítmény

    Függvények:
        - power_by_wheel(loss_percent)
        - get_power()

    Eljárások:
        - write_out_wheel_number()
        - write_out_power()
        - set_power(int kw)

```
CREATE OR REPLACE TYPE VEHICLE_O AS OBJECT (
                NUMBER_OF_WHEELS NUMBER(8,5),
                POWER NUMBER(8,5),
                MEMBER FUNCTION POWER_BY_WHEEL (LOSS_PERCENT
NUMBER)    RETURN NUMBER,
                MEMBER FUNCTION GET_POWER RETURN NUMBER,
                MEMBER PROCEDURE WRITE_OUT_WHEEL_NUMBER,
                MEMBER PROCEDURE WRITE_OUT_POWER,
                MEMBER PROCEDURE SET_POWER (P NUMBER)
) NOT FINAL;
```



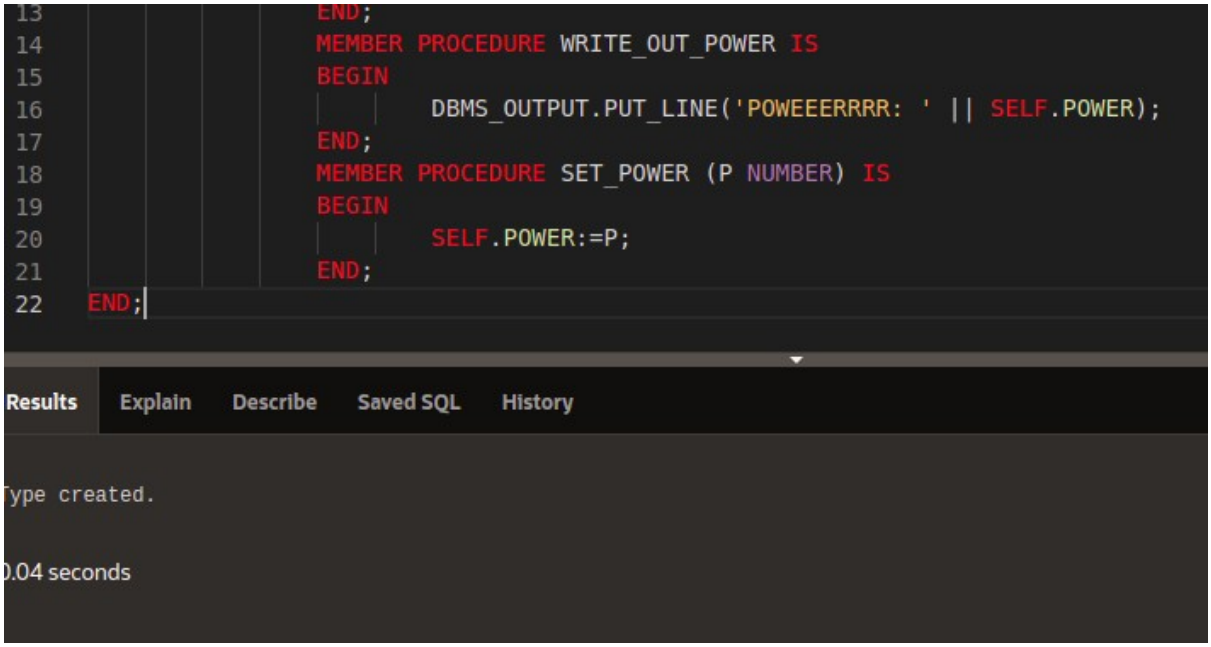

```
CREATE OR REPLACE TYPE BODY VEHICLE_O IS
                MEMBER FUNCTION POWER_BY_WHEEL(LOSS_PERCENT
NUMBER) RETURN NUMBER IS
                BEGIN
                        RETURN (SELF.POWER / SELF.NUMBER_OF_WHEELS)*
(1 - LOSS_PERCENT / 100);
                END;
                MEMBER FUNCTION GET_POWER RETURN NUMBER IS
                BEGIN
```

```
                RETURN SELF.POWER;
        END;
        MEMBER PROCEDURE WRITE_OUT_WHEEL_NUMBER IS
        BEGIN
                DBMS_OUTPUT.PUT_LINE('NUMBER OF WHEELS: ' ||
SELF.NUMBER_OF_WHEELS);
        END;
        MEMBER PROCEDURE WRITE_OUT_POWER IS
        BEGIN
                DBMS_OUTPUT.PUT_LINE('POWEEERRRR: ' ||
SELF.POWER);
        END;
        MEMBER PROCEDURE SET_POWER (P NUMBER) IS
        BEGIN
                SELF.POWER:=P;
        END;
END;
```

b. Hozzunk létre egy car_o objektumot a vehicle_o osztály gyerekeként,
   Adattagok:
   - registration_number
   - color
   - production_year
   - manufacturer
   - price

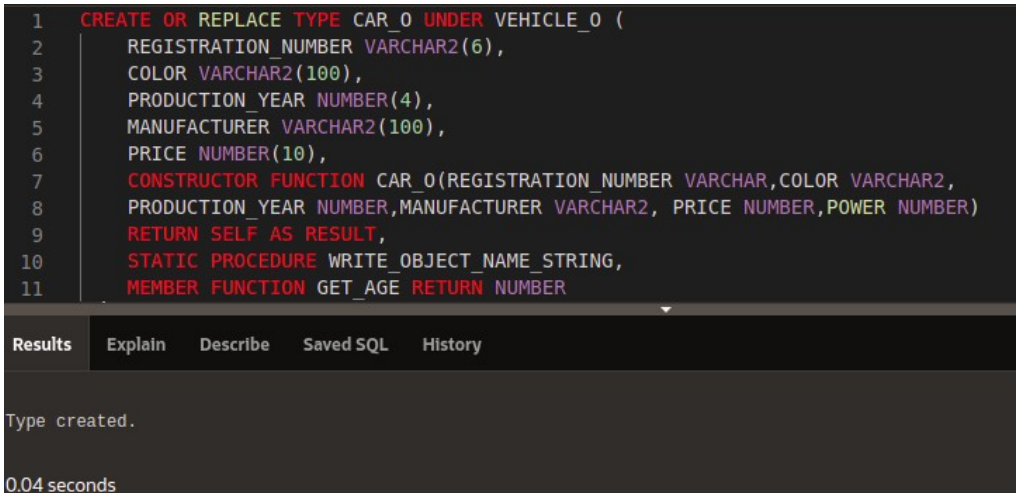
   Statikus eljárás:
   - write_object_name_sring()

   Függvény:
   - get_age()



   Legyen konstruktor!

```
CREATE OR REPLACE TYPE CAR_O UNDER VEHICLE_O (
   REGISTRATION_NUMBER VARCHAR2(6),
   COLOR VARCHAR2(100),
   PRODUCTION_YEAR NUMBER(4),
   MANUFACTURER VARCHAR2(100),
   PRICE NUMBER(10),
   CONSTRUCTOR FUNCTION CAR_O(REGISTRATION_NUMBER
VARCHAR,COLOR VARCHAR2,
   PRODUCTION_YEAR NUMBER,MANUFACTURER VARCHAR2, PRICE
NUMBER,POWER NUMBER)
   RETURN SELF AS RESULT,
   STATIC PROCEDURE WRITE_OBJECT_NAME_STRING,
   MEMBER FUNCTION GET_AGE RETURN NUMBER
);
```




```
CREATE OR REPLACE TYPE BODY CAR_O AS
   CONSTRUCTOR FUNCTION CAR_O(REGISTRATION_NUMBER
VARCHAR,COLOR VARCHAR2,
   PRODUCTION_YEAR NUMBER,MANUFACTURER VARCHAR2, PRICE
NUMBER,POWER NUMBER)
   RETURN SELF AS RESULT
   AS
   BEGIN
      SELF.REGISTRATION_NUMBER := REGISTRATION_NUMBER;
      SELF.COLOR := COLOR;
      SELF.PRODUCTION_YEAR := PRODUCTION_YEAR;
      SELF.MANUFACTURER := MANUFACTURER;
      SELF.PRICE := PRICE;
      SELF.POWER := POWER;
      SELF.NUMBER_OF_WHEELS := 4;
      RETURN;
   END;
```

```
    STATIC PROCEDURE WRITE_OBJECT_NAME_STRING AS
    BEGIN
      DBMS_OUTPUT.PUT_LINE('STATIC PROCEDURE');
    END;
    MEMBER FUNCTION GET_AGE RETURN NUMBER AS
    BEGIN
      RETURN TO_CHAR(SYSDATE,'YYYY') - SELF.PRODUCTION_YEAR;
    END;
END;
```

```
CREATE OR REPLACE TYPE BODY CAR_O AS
    CONSTRUCTOR FUNCTION CAR_O(REGISTRATION_NUMBER VARCHAR,COLOR VARCHAR2,
    PRODUCTION_YEAR NUMBER,MANUFACTURER VARCHAR2, PRICE NUMBER,POWER NUMBER)
    RETURN SELF AS RESULT
    AS
    BEGIN
        SELF.REGISTRATION_NUMBER := REGISTRATION_NUMBER;
        SELF.COLOR := COLOR;
        SELF.PRODUCTION_YEAR := PRODUCTION_YEAR;
        SELF.MANUFACTURER := MANUFACTURER;
        SELF.PRICE := PRICE;
```

Results | Explain | Describe | Saved SQL | History

```
Type created.
```

c. Próbáljuk ki egy névtelen plsql blokkon belül a létrehozott típus minden függvényét és eljárását.

```
DECLARE
 car CAR_O;
BEGIN
   CAR := CAR_O('AAA111','zöld',2010,'opel',2000000,450);
   CAR_O.WRITE_OBJECT_NAME_STRING();
   CAR.WRITE_OUT_WHEEL_NUMBER();
   CAR.WRITE_OUT_POWER();
   DBMS_OUTPUT.PUT_LINE('KOR: ' || CAR.GET_AGE());
   DBMS_OUTPUT.PUT_LINE('POWER BY WHEEL: ' ||
CAR.POWER_BY_WHEEL(10));
   CAR.SET_POWER(10);
   DBMS_OUTPUT.PUT_LINE('POWER: ' || CAR.GET_POWER());
END;
```

```
3   BEGIN
4       CAR := CAR_O('AAA111','zöld',2010,'opel',2000000,450);
5       CAR_O.WRITE_OBJECT_NAME_STRING();
6       CAR.WRITE_OUT_WHEEL_NUMBER();
7       CAR.WRITE_OUT_POWER();
8       DBMS_OUTPUT.PUT_LINE('KOR: ' || CAR.GET_AGE());
9       DBMS_OUTPUT.PUT_LINE('POWER BY WHEEL: ' || CAR.POWER_BY_WHEEL(10));
10      CAR.SET_POWER(10);
11      DBMS_OUTPUT.PUT_LINE('POWER: ' || CAR.GET_POWER());
12  END;
```

Results | Explain | Describe | Saved SQL | History

```
STATIC PROCEDURE
NUMBER OF WHEELS: 4
POWEEERRRR: 450
KOR: 12
POWER BY WHEEL: 101.25
POWER: 10
```