**JEGYZŐKÖNYV**

Modern adatbázis rendszerek Msc

2022. tavasz féléves feladat

Oracle ORDBMS

Készítette: **Polonkai Dávid**

Neptunkód: **GPNWZT**

**Modern Adatbázis rendszerek hallgatói jegyzőkönyv**

1. Car objektum
   1. Hozz létre egy car objektumot (registration\_number, color, production\_year, manufacturer, price) attribútumokkal.
   2. Hozz létre egy táblát, amelynek neve car\_showroom (car\_id, car)
   3. Adj hozzá adatokat
   4. Kérdezd le minden car objektum színét a car\_showroom táblából
   5. Állítsd be, hogy minden car fehér legyen ami 5 évnél idősebb
   6. (szorgalmi) \* Oldjuk meg ezt member procedure-val \*
2. Nested table
   1. Hozd létre a car\_table\_type tábla típust, ami car-okat tartalmaz
   2. Hozd létre a showroom\_network táblát, ami (id, city, car\_table) adattagokat  
      tartalmaz
   3. Vigyél fel egy ‘opel’-t és ‘skoda’-t a Miskolc városához tartozó táblába.
   4. Vigyél fel egy ‘opel’-t a Debrecen városához tartozó táblába.
   5. Kérdezzük le a Miskolci szalonban található autókat
   6. Kérdezzük le minden adatot, figyeljük meg a kimenetét
   7. Kérdezzük le a nested tábla adatait és figyeljük meg mit találunk
3. OOP
   1. 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)
  1. 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!**

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

**Modern Adatbázis rendszerek gyakorlatvezetői jegyzőkönyv**

1. Car objektum
   1. Hozz létre egy car objektumot (registration\_number, color, production\_year, manufacturer, price) attribútumokkal.

CREATE TYPE CAR\_T AS OBJECT(

REGISTRATION\_NUMBER VARCHAR2(100),

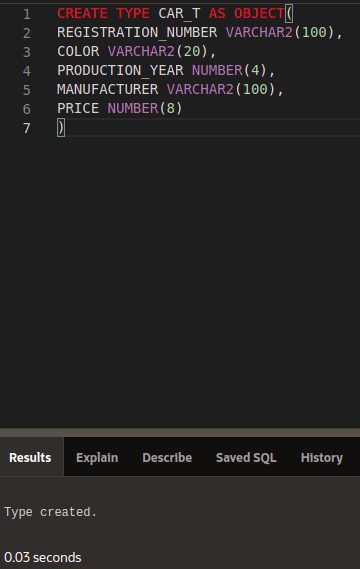
COLOR VARCHAR2(20),

PRODUCTION\_YEAR NUMBER(4),

MANUFACTURER VARCHAR2(100),

PRICE NUMBER(8)

)



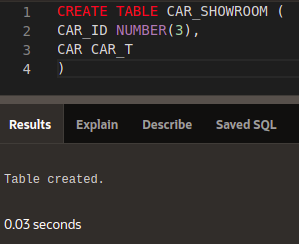
* 1. Hozz létre egy táblát, amelynek neve car\_showroom (car\_id, car)

CREATE TABLE CAR\_SHOWROOM (

CAR\_ID NUMBER(3),

CAR CAR\_T

)

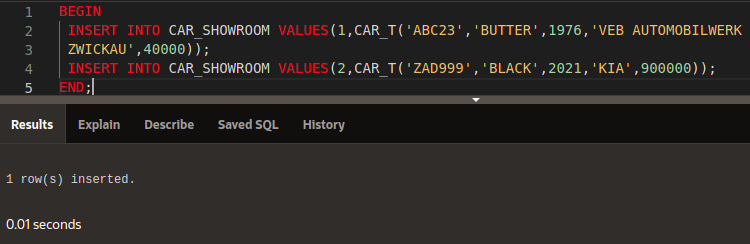


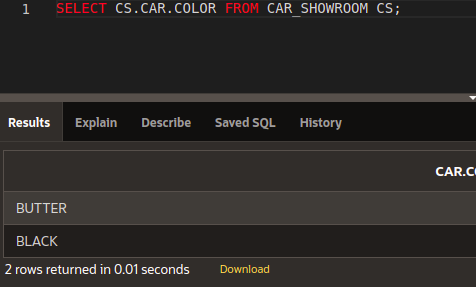
* 1. Adj hozzá adatokat

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));

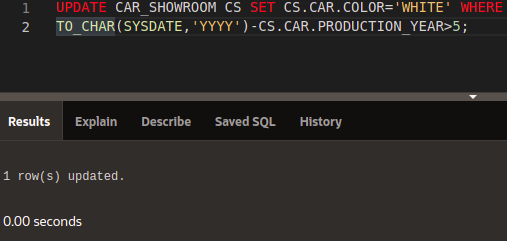
(A nevesítetlen plsql blokk miatt csak az utolsó művelet kimenetét írja ki a consolera az oracle szerver)

* 1. Kérdezd le minden car objektum színét a car\_showroom táblából

SELECT CS.CAR.COLOR FROM CAR\_SHOWROOM CS;

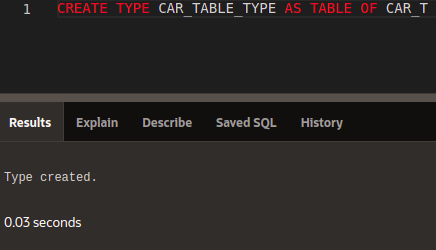
* 1. Állítsd be, hogy minden car fehér legyen ami 5 évnél idősebb

UPDATE CAR\_SHOWROOM CS SET CS.CAR.COLOR='WHITE' WHERE TO\_CHAR(SYSDATE,'YYYY')-CS.CAR.PRODUCTION\_YEAR>5;



1. Nested table
   1. Hozd létre a car\_table\_type tábla típust, ami car-okat tartalmaz

CREATE TYPE CAR\_TABLE\_TYPE AS TABLE OF CAR\_T



* 1. Hozd létre a showroom\_network táblát, ami (id, city, car\_table) adattagokat  
     tartalmaz

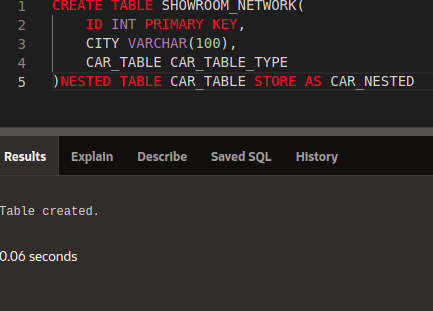
CREATE TABLE SHOWROOM\_NETWORK(

ID INT PRIMARY KEY,

CITY VARCHAR(100),

CAR\_TABLE CAR\_TABLE\_TYPE

)NESTED TABLE CAR\_TABLE STORE AS CAR\_NESTED



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

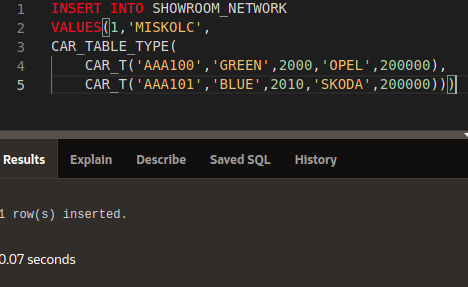
INSERT INTO SHOWROOM\_NETWORK

VALUES(1,'MISKOLC',

CAR\_TABLE\_TYPE(

CAR\_T('AAA100','GREEN',2000,'OPEL',200000),

CAR\_T('AAA101','BLUE',2010,'SKODA',200000)))



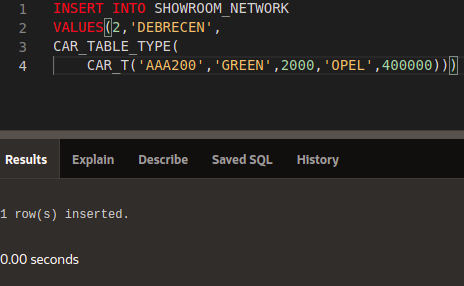
* 1. Vigyél fel egy ‘opel’-t a Debrecen városához tartozó táblába.

INSERT INTO SHOWROOM\_NETWORK

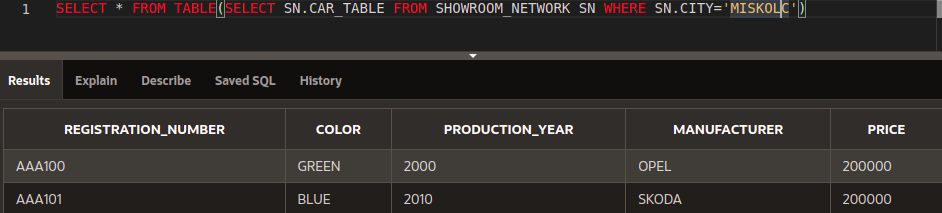
VALUES(2,'DEBRECEN',

CAR\_TABLE\_TYPE(

CAR\_T('AAA200','GREEN',2000,'OPEL',400000)))

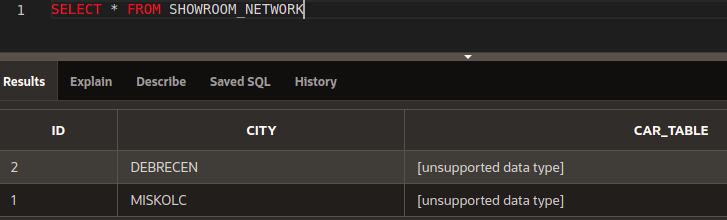
* 1. Kérdezzük le a Miskolci szalonban található autókat

SELECT \* FROM TABLE(SELECT SN.CAR\_TABLE FROM SHOWROOM\_NETWORK SN WHERE SN.CITY='MISKOLC')



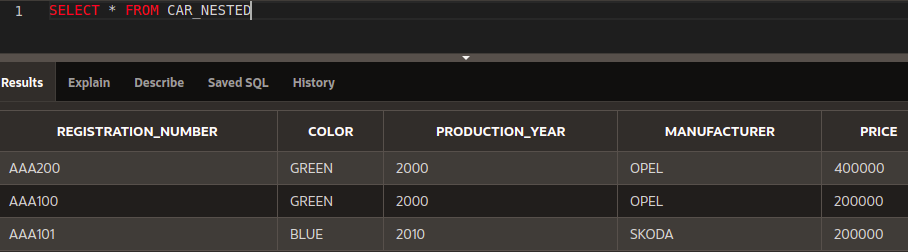
* 1. Kérdezzük le minden adatot, figyeljük meg a kimenetét

SELECT \* FROM SHOWROOM\_NETWORK



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

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

SELECT \* FROM CAR\_NESTED

1. OOP
   1. 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,

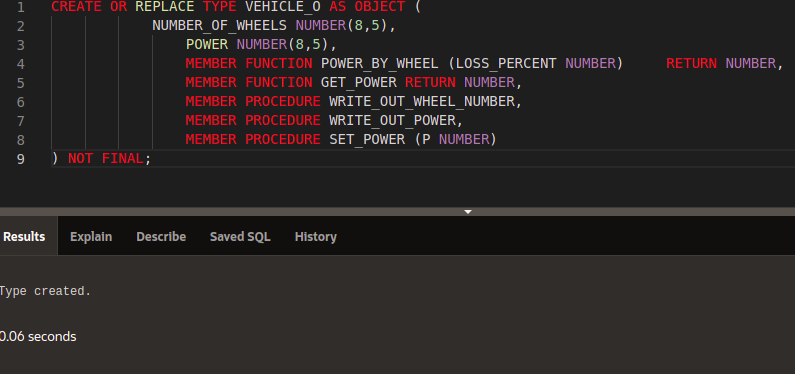
MEMBER FUNCTION GET\_POWER RETURN NUMBER,

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;

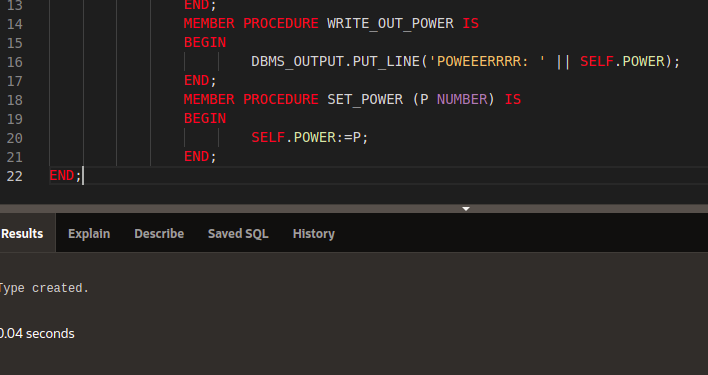
MEMBER PROCEDURE SET\_POWER (P NUMBER) IS

BEGIN

SELF.POWER:=P;

END;

END;



* 1. 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,

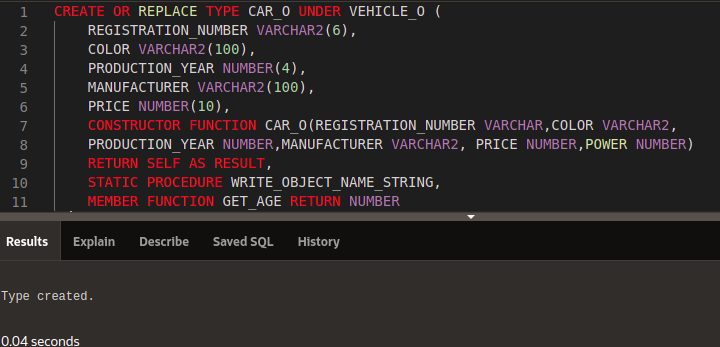
PRODUCTION\_YEAR NUMBER,MANUFACTURER VARCHAR2, PRICE NUMBER,POWER NUMBER)

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;

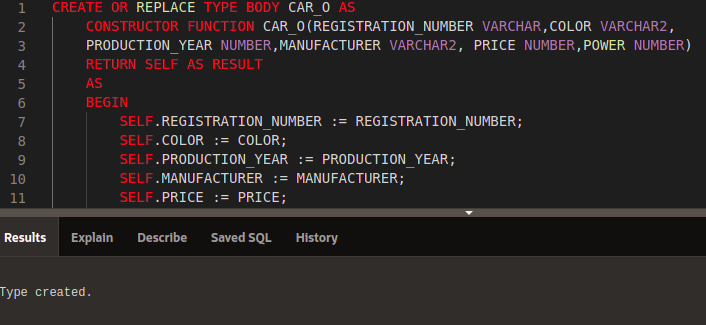
MEMBER FUNCTION GET\_AGE RETURN NUMBER AS

BEGIN

RETURN TO\_CHAR(SYSDATE,'YYYY') - SELF.PRODUCTION\_YEAR;

END;

END;



* 1. 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;

