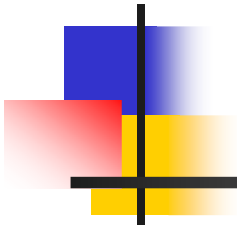
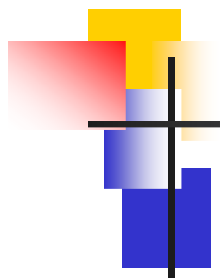


CURS 5



Componenta obiect – relațională în Oracle (II) Colecții



Colecții

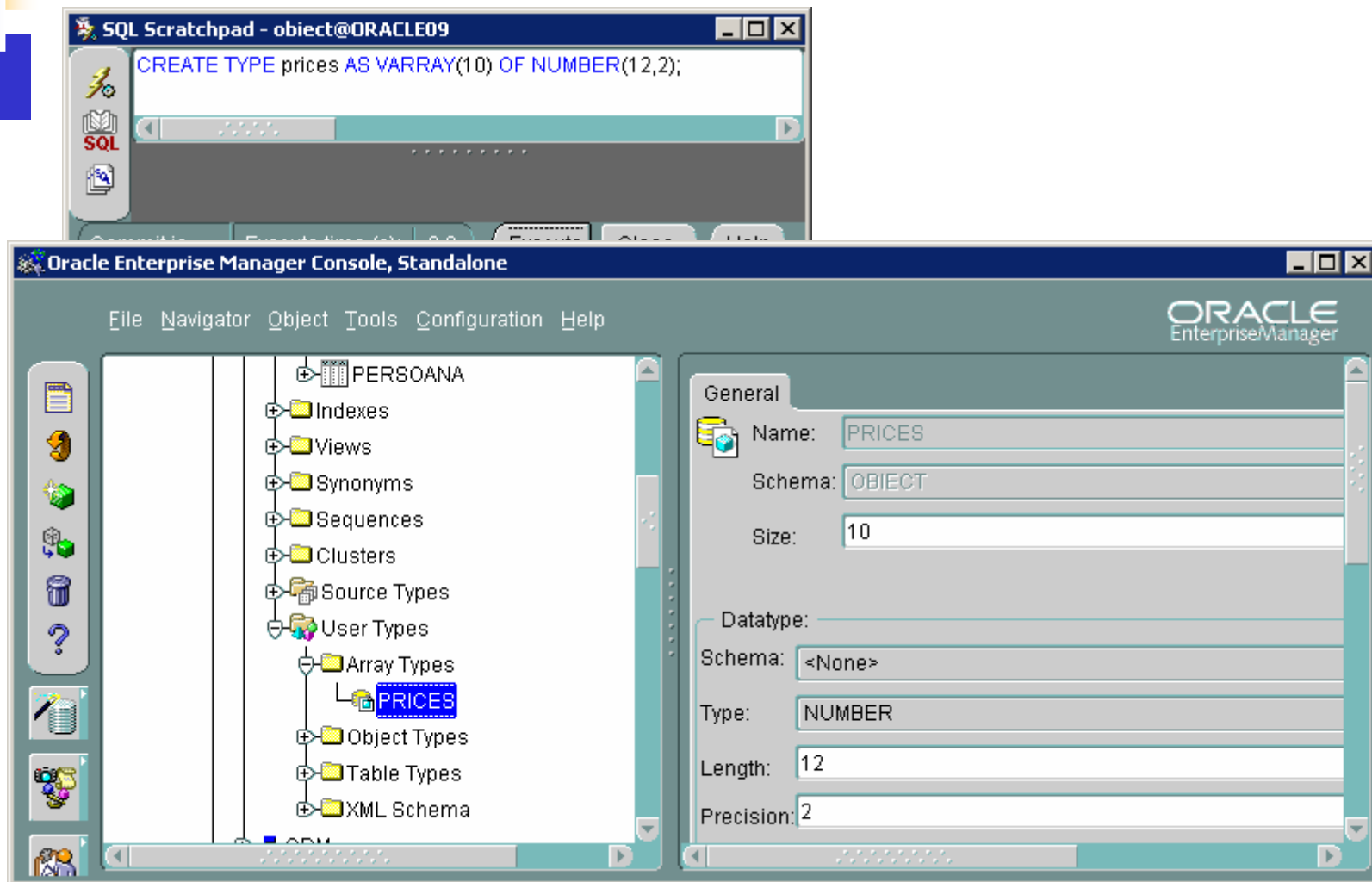
- Se folosesc pentru modelarea legăturilor 1:N
 - Vectori
 - Tabele încapsulate
- La crearea unui tip colecție nu se alocă spațiu, ci doar se definește un tip de dată care poate fi folosit ca:
 - tipul datei dintr-o coloană a unui tabel relațional
 - atribut pentru un tip obiect
 - tipul unei variabile PL/SQL, ca parametru sau ca valoare returnată de o funcție

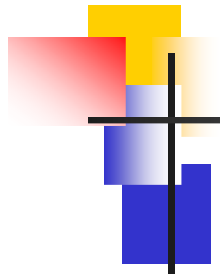


Date de tip vector - array

- Un vector este o mulțime ordonată de elemente, de același tip. Fiecare element are un index ce corespunde poziției elementului în vector
- Numărul elementelor reprezintă dimensiunea vectorului.
 - Oracle permite ca vectorii să aibă dimensiuni variabile → *varrays*.
 - La declararea tipului *array* trebuie specificată dimensiunea maximă pe care o poate avea un vector.
- În mod obișnuit, un vector este stocat în linie, și ca urmare, în același tablespace ca celelalte date din rândul său. Dacă este suficient de mare, Oracle îl memorează ca BLOB.
- Un vector nu poate conține date de tip LOB. Aceasta înseamnă că un vector nu poate conține elemente ale unui tip de date definit de utilizator care are un atribut de tip LOB.

Exemplu





Tabele încapsulate (nested tables)

- Un tabel încapsulat este o mulțime neordonată de elemente , toate de același tip.
- Are o singură coloană și tipul acesteia este un tip predefinit sau un tip obiect.
 - În ultimul caz tabelul poate fi văzut ca un tabel multicoloană, câte o coloană pentru fiecare atribut al tipului obiect..
- Când o coloană dintr-un tabel relațional este de tip *nested table*, Oracle memorează datele din tabelul încapsulat pentru toate rândurile tabelului relațional în același tabel.
- Similar cu obiectul tabel creat pe un tip de dată care are un atribut de tip *nested table*, Oracle memorează datele din tabelul încapsulat pentru toate instanțele obiectului într-un singur tabel asociat cu obiectul tabel care conține datele.

Exemplu înregistrarea comenzilor unei/către o firmă

Formular comandă

Nr. Comanda.		
Data.		
codart	denumire	cantitate
1	Articol 1	Cant 1
2	Articol 2	Cant 2
....		

Tabel comenzi

Nr. Comanda	Data	Linii articole		
		codart	denumre	cantitate
1	D1	Art1	Den1	C1
		Art2	Den 2	C2
2	D2	Art 1	Den 1	C3
		Art 3	Den 3	C4
		Art 4	Den 4	C5

Ce soluții avem ????

Exemplu înregistrarea comenzilor unei/către o firmă

Formular comandă

Nr. Comanda. Data.		
codart	denumire	cantitate
1	Articol 1	Cant 1
2	Articol 2	Cant 2
....		

Tip de dată
linie_art

Obiect tabel de
tipul *linie_art*

```
SQL Scratchpad - obiect@ORACLE09

create type linie_art AS Object (
codart varchar2(3),
denumire varchar2(25),
cantitate number(5,2));

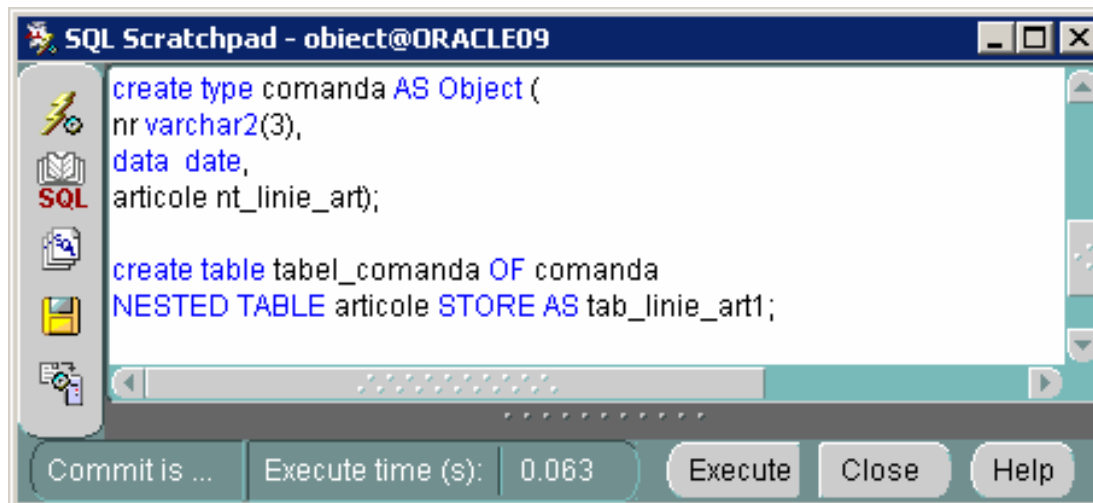
CREATE TYPE nt_linie_art AS TABLE OF linie_art;
```

Commit is ... Execute time (s): 0.063 Execute Close Help

Exemplu înregistrarea comenzilor unei/către o firmă

Tabel comenzi

Nr. Comanda	Data	Linii articole		
		codart	denumre	cantitate
1	D1	Art1	Den1	C1
		Art2	Den 2	C2
2	D2	Art 1	Den 1	C3
		Art 3	Den 3	C4
		Art 4	Den 4	C5

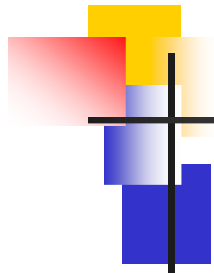


```
SQL Scratchpad - obiect@ORACLE09

create type comanda AS Object (
  nr varchar2(3),
  data date,
  articole nt_linie_art);

create table tabel_comanda OF comanda
  NESTED TABLE articole STORE AS tab_linie_art1;
```

Commit is ... Execute time (s): 0.063 Execute Close Help



Oracle Enterprise Manager Console, Standalone

File Navigator Object Tools Configuration Help

ORACLE
Enterprise Manager

OEMREP
ORACLE09 - object
Instance
Schema
CTXSYS
HR
MDSYS
OBJECT
Tables
ADR_TAB
ANG_TABLE
PERSOANA
TABEL_COMANDA
TAB_LINIE_ART
Indexes
Views
Synonyms
Sequences
Clusters

General Constraints Storage Statistics

Name: TABEL_COMANDA
Schema: OBJECT
Tablespace: USERS

Table: ☒ Standard ☐ Organized Using Index (IOT) ☐ Use Abstract Datatype

Object Table Attributes
Schema: OBJECT
Object Type: COMANDA
Nested Tables...

Name	Datatype	Null...	Default Value	Scope	Schema	T
NR	VARCHAR2(3)	✓				
DATA	DATE	✓				
ARTICOLE	OBJECT.NT_LINIE_ART	✓				

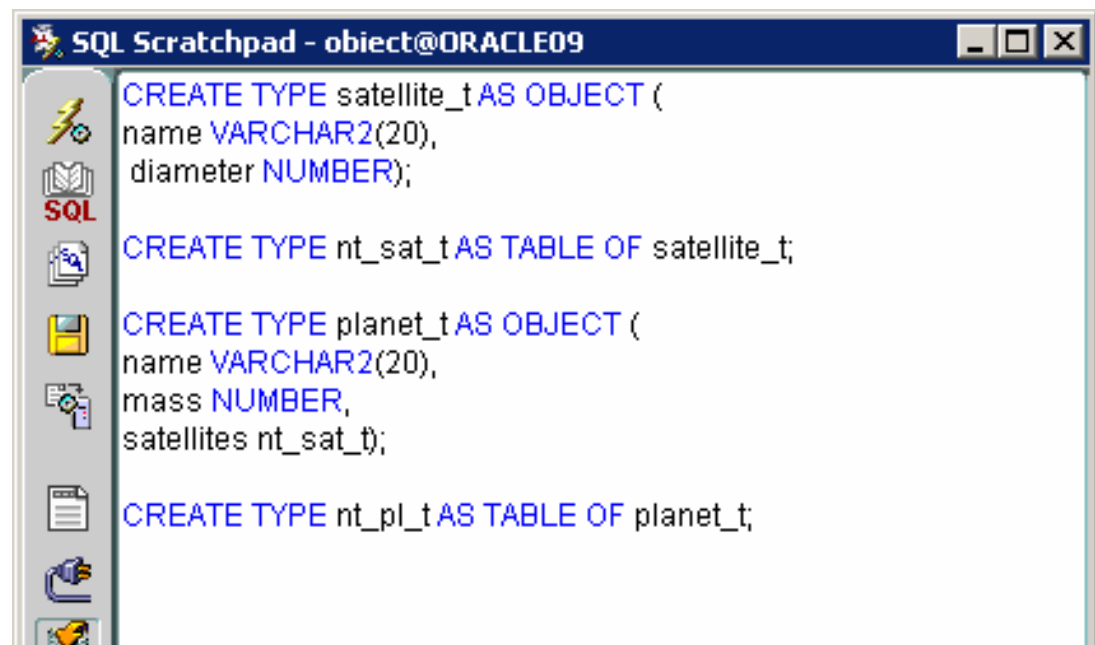


Colecții pe mai multe nivele

- Tipurile colecție multinivel sunt tipuri colecție ale căror elemente sunt ele însele direct sau indirect alte tipuri colecție.
- Cazuri posibile:
 - Tabel încapsulat de tip tabel încapsulat
 - Tabel încapsulat de tip vector variabil
 - Vector variabil de tip tabel încapsulat
 - Vector variabil de tip vector variabil
 - Tabel încapsulat sau vector variabil de tip definit de utilizator care are un atribut care este de tip tabel încapsulat sau vector variabil
- Ca și tipurile colecție pe un singur nivel tipurile de colecții multi-nivel pot fi utilizate ca și coloane într-un tabel relațional sau ca și attribute obiect într-un obiect tabel.

Exemplu

- Se creează un tip de colecție multinivel care este reprezentat de un tabel încapsulat de tabele încapsulate
- Se modelează un sistem stelar în care fiecare stea are o colecție de planete care se rotesc în jurul său(tabel încapsulat) și fiecare planetă are o serie de sateliți stocați într-un tabel încapsulat.



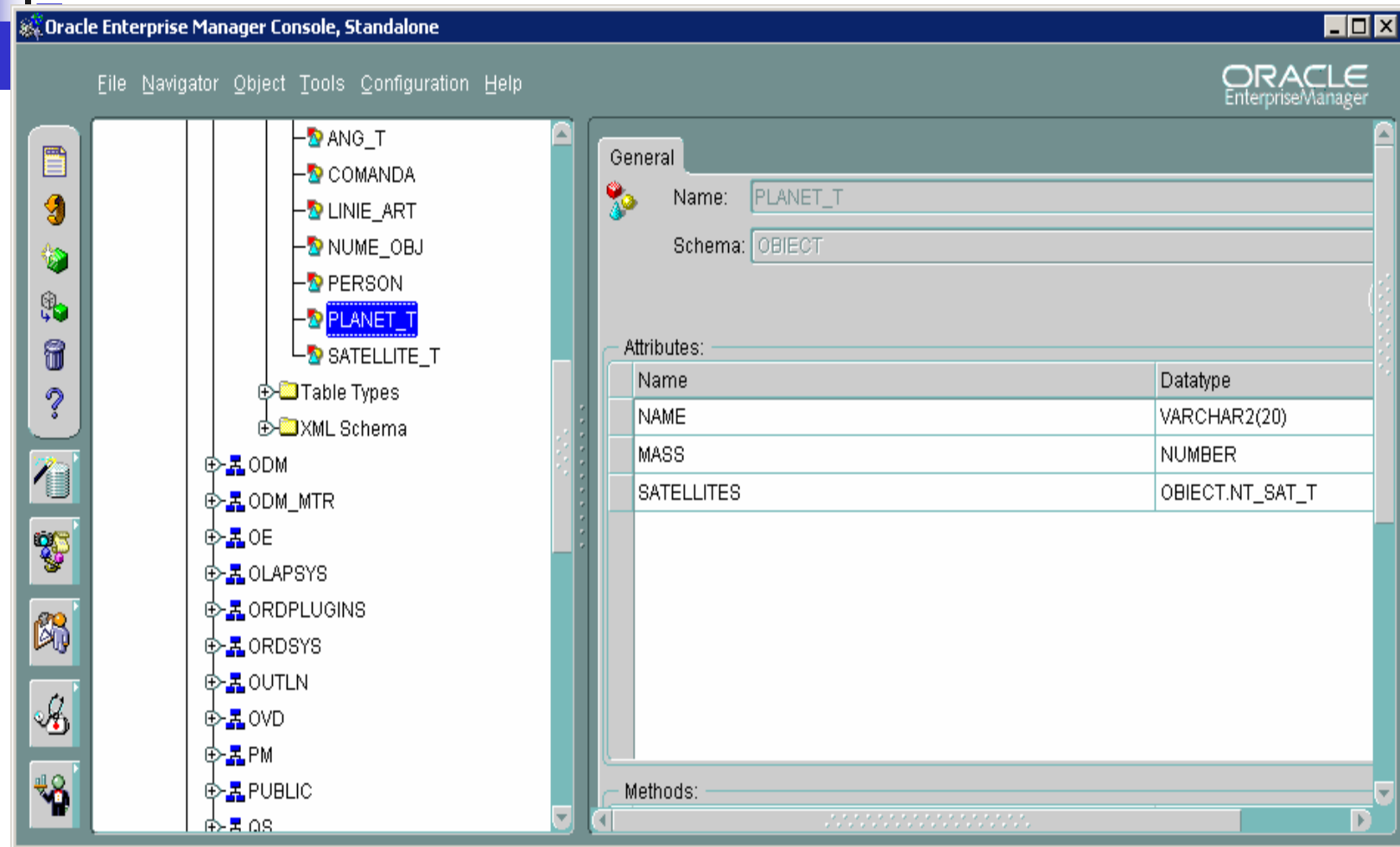
```
SQL Scratchpad - obiect@ORACLE09

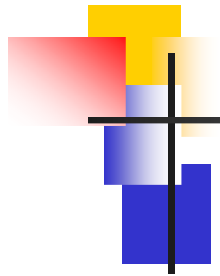
CREATE TYPE satellite_t AS OBJECT (
  name VARCHAR2(20),
  diameter NUMBER);

CREATE TYPE nt_sat_t AS TABLE OF satellite_t;

CREATE TYPE planet_t AS OBJECT (
  name VARCHAR2(20),
  mass NUMBER,
  satellites nt_sat_t);

CREATE TYPE nt_pl_t AS TABLE OF planet_t;
```





Stocarea tabelelor încapsulate

- Un atribut (aparținând unui obiect tabel) sau o coloană de tip tabel încapsulat necesită un tabel (storage table) care stochează rândurile aparținând tuturor tabelelor din coloană.
- Similar unei colecții multinivel, de tabele încapsulate în tabele încapsulate, mulțimea interioară de tabele solicită un tabel de stocare ca și mulțimea exterioară.
 - Acesta poate fi specificat prin adăugarea unei clauze suplimentare pentru stocarea tabelului încapsulat.

Stocarea tabelelor încapsulate

DATA1	DATA2	DATA3	DATA4	NT_DATA
...	A
...	B
...	C
...	D
...	E

- Oracle stochează rândurile unui tabel încapsulat într-un tabel separat.
- Sistemul generează un câmp NESTED_TABLE_ID, pe 16 octeți care corelează rândurile părinte cu rândurile corespondente din tabelul de stocare.

Storage Table

NESTED_TABLE_ID	Values
B	B21
B	B22
C	C33
A	A11
E	E51
B	B25
E	E52
A	A12
E	E54
B	B23
C	C32
A	A13
D	D41
B	B24
E	E53

Tabele încapsulate stocate în tabele indexate (IOT- Index-Organized Table)

- Dacă un tabel încapsulat are o cheie primară, atunci acesta poate fi organizat ca un tabel indexat (IOT-index-organized table).
- Dacă coloana NESTED_TABLE_ID este prefix al cheii primare pentru un rând al tabelului părinte, Oracle grupează fizic rândurile copil.
- Deci când este accesat un rând părinte pot fi regăsite eficient toate rândurile copil asociate.
- Chiar dacă se accesează numai rândurile părinte eficiența este crescută deoarece acestea nu sunt inter-mixate cu rândurile copil.

DATA1	DATA2	DATA3	DATA4	NT_DATA
...	A
...	B
...	C
...	D
...	E

Storage Table

NESTED_TABLE_ID	Values
A	A11
A	A12
A	A13
B	B21
B	B22
B	B23
B	B24
B	B25
C	C31
C	C32
D	D41
E	E51
E	E52
E	E53
E	E54

Storage for nested table A

Storage for nested table B

Storage for nested table C

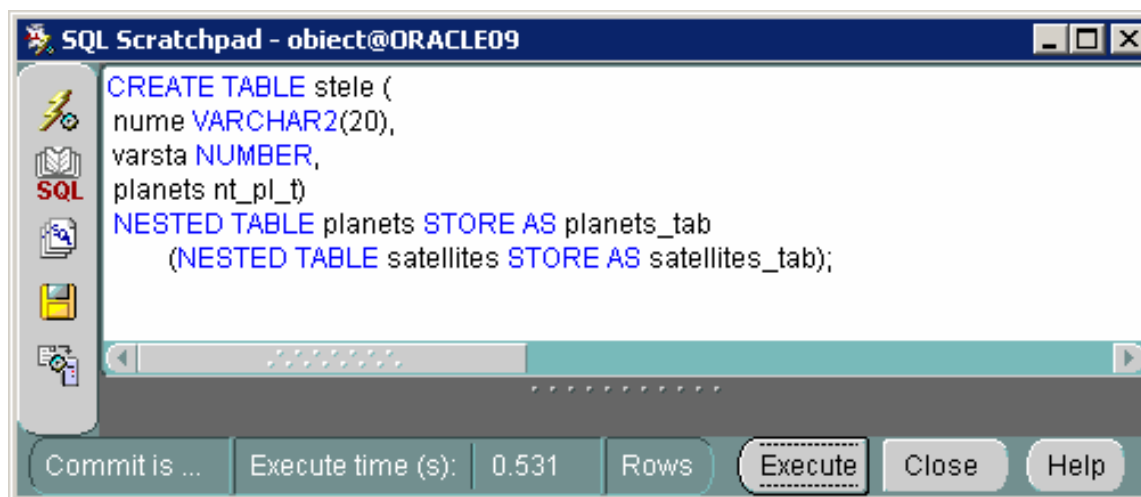
Storage for nested table D

Storage for nested table E

Tabele în care sunt stocate tabele încapsulate- Exemplu 1

Se crează un tabel *Stele* care conține o coloana *planets* al cărui tip este o colecție multinivel (un tabel încapsulat al cărui tip are un atribut *satellites* de tip tabel încapsulat).

- Se furnizează clauze de stocare separate atât pentru tabelul încapsulat exterior planete cât și pentru cel interior satelit.



The screenshot shows a window titled "SQL Scratchpad - obiect@ORACLE09". The main text area contains the following SQL code:

```
CREATE TABLE stele (  
  nume VARCHAR2(20),  
  varsta NUMBER,  
  planets nt_pl_t)  
  NESTED TABLE planets STORE AS planets_tab  
    (NESTED TABLE satellites STORE AS satellites_tab);
```

At the bottom of the window, there is a status bar with the following information: "Commit is ...", "Execute time (s): 0.531", "Rows", and buttons for "Execute", "Close", and "Help".

- In acest caz se poate referi tabelul interior *satellite* prin nume deoarece acest tabel este un atribut cu nume într-un obiect.

Tabele în care sunt stocate tabele încapsulate- Exemplu 2

- Dacă tabelul încapsulat interior nu este atribut el nu are nume. In acest caz există cuvântul cheie `COLUMN_VALUE` care se poate folosi în locul numelui unui tabel încapsulat interior.

The screenshot displays the Oracle Enterprise Manager interface and a SQL Scratchpad window. The SQL Scratchpad window shows the following SQL code:

```
CREATE TYPE inner_table AS TABLE OF NUMBER;  
CREATE TYPE outer_table AS TABLE OF inner_table;  
CREATE TABLE tab1 (  
  col1 NUMBER,  
  col2 outer_table)  
NESTED TABLE col2 STORE AS col2_ntab  
(NESTED TABLE COLUMN_VALUE STORE AS cv_ntab);
```

The Oracle Enterprise Manager interface shows the structure of the table `TAB1` in the `OBJECT` schema, located in the `USERS` tablespace. The table is defined as a Standard table. The columns are:

Name	Schema	Datatype	Size	Scale	Ref	Nulls?	Default Value
COL1	<None>	NUMBER				✓	
COL2	OBJECT	OUTER_TABLE				✓	



Stocarea datelor multinivel de tip varray

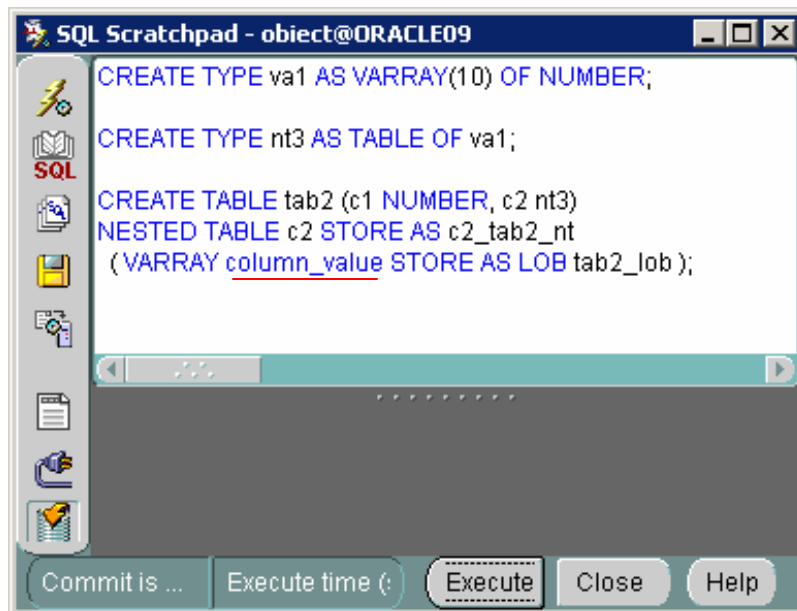
Există două posibilități de stocare funcție de tipul colecției multinivel:

1. Vector de vectori variabili
 2. Vector de tabele încapsulate.
1. Întregul vector este stocat în linie dacă acesta nu este prea mare (până în 4000 octeți) sau dacă nu este specificat explicit opțiunea de stocare LOB
 2. Întregul vector este stocat într-un LOB și în rând se stochează numai locatorul pentru LOB
 1. Nu există tabele de stocare asociate cu elementele de tabel încapsulat al unui vector.
 2. Întreaga colecție este memorată în interiorul vectorului.
- Se poate specifica explicit opțiunea de stocare LOB pentru vectori variabili.

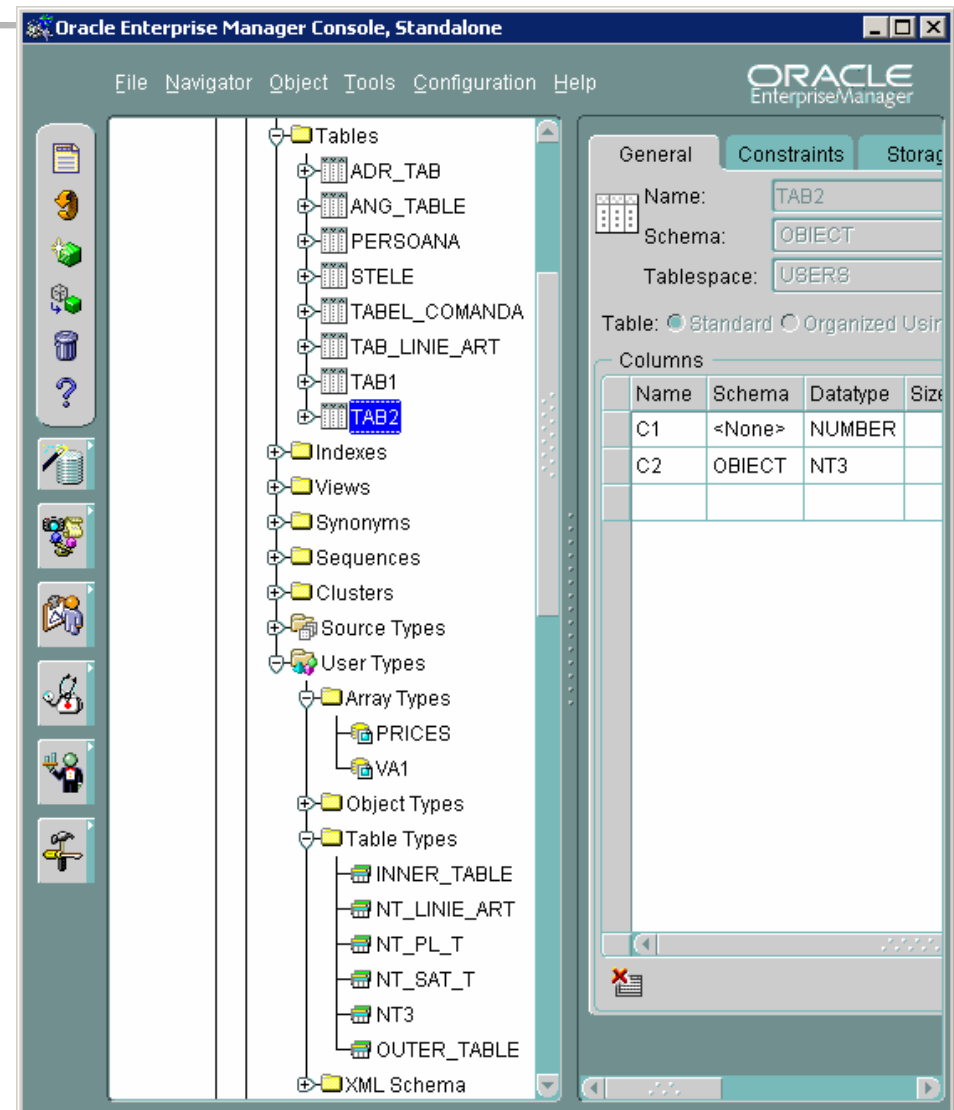
Stocarea datelor multinivel de tip varray

Se consideră cazul elementelor varray ale unui tabel încapsulat.

Se prezintă modul de utilizare al cuvântului cheie **COLUMN_VALUE** împreună cu tipul varray.

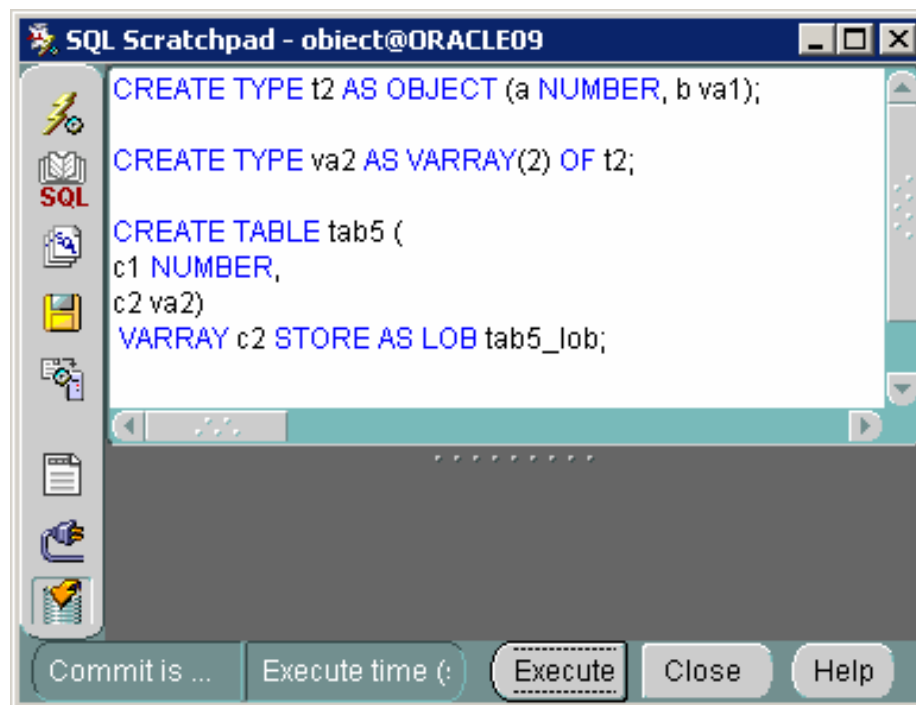


```
CREATE TYPE va1 AS VARRAY(10) OF NUMBER;  
CREATE TYPE nt3 AS TABLE OF va1;  
CREATE TABLE tab2 (c1 NUMBER, c2 nt3)  
NESTED TABLE c2 STORE AS c2_tab2_nt  
( VARRAY column_value STORE AS LOB tab2_lob );
```



Stocarea datelor multinivel de tip varray

În exemplu se arată modul de specificare explicită ca stocare LOB pentru un vector de vectori variabili:



The screenshot shows a window titled "SQL Scratchpad - obiect@ORACLE09". The window contains the following SQL code:

```
CREATE TYPE t2 AS OBJECT (a NUMBER, b va1);  
CREATE TYPE va2 AS VARRAY(2) OF t2;  
CREATE TABLE tab5 (  
  c1 NUMBER,  
  c2 va2)  
  VARRAY c2 STORE AS LOB tab5_lob;
```

The window has a toolbar on the left with icons for SQL, file operations, and execution. At the bottom, there are buttons for "Commit is ...", "Execute time (:)", "Execute", "Close", and "Help".



Crearea obiectelor VARRAY sau Nested Table

Instanța unui tip colecție se creează în același mod ca instanța oricărui alt tip obiect, și anume, prin apelarea metodei constructor corespunzătoare

- numele acesteia este chiar numele tipului
- Elementele colecției se dau ca o listă de argumente, separate prin virgulă, pentru metoda apelată
- Apelarea unui constructor cu o listă vidă de argumente creează o colecție vidă (*empty collection*) de tipul respectiv.
 - O colecție vidă este o colecție care (posibil din întâmplare) în prezent este goală.
 - NU are aceeași semnificație cu o colecție *null*.



Constructori pentru colecții multinivel

- Tipurile de colecții multinivel sunt create prin apelarea metodei constructor aferentă tipului
 - Constructorul unui tip de colecție multinivel este o funcție definită de sistem care are același nume cu tipul respectiv și întoarce o instanță nouă a acestuia.
 - Parametrii transmiși constructorului au numele și tipurile corespunzătoare tipurilor atributelor obiectului.
- Exemplu: colecție de tip tabel încapsulat de tabele încapsulate
 - Constructorul pentru tabelul încapsulat exterior apelează constructorul *planet_t* pentru fiecare *planeta* ce se creează
 - Constructorul fiecărei planete apelează constructorul pentru tabelul încapsulat *satellites* pentru a crea tabelul *satelittes*
 - Constructorul tipului *satellites* (tabel încapsulat) apelează constructorul *satellite_t* pentru a crea instanțe pentru fiecare satelit

SQL Scratchpad - obiect@ORACLE09

```
INSERT INTO stele VALUES(  
  'Sun',23,  
  nt_pl_t(  
    planet_t(  
      'Neptune', 10,  
      nt_sat_t(  
        satellite_t(  
          'Proteus',67),  
        satellite_t(  
          'Triton',82) ) ),  
    planet_t(  
      'Jupiter', 189,  
      nt_sat_t(  
        satellite_t(  
          'Callisto',97),  
        satellite_t(  
          'Ganymede', 22) ) ) ) );
```

Commit is ... Execute time (s): 0.1 Execute Close Help

Oracle SQL*Plus

Fișier Editare Căutare Opțiuni Asistență

```
SQL> select * from stele;
```

NUME	VARSTA

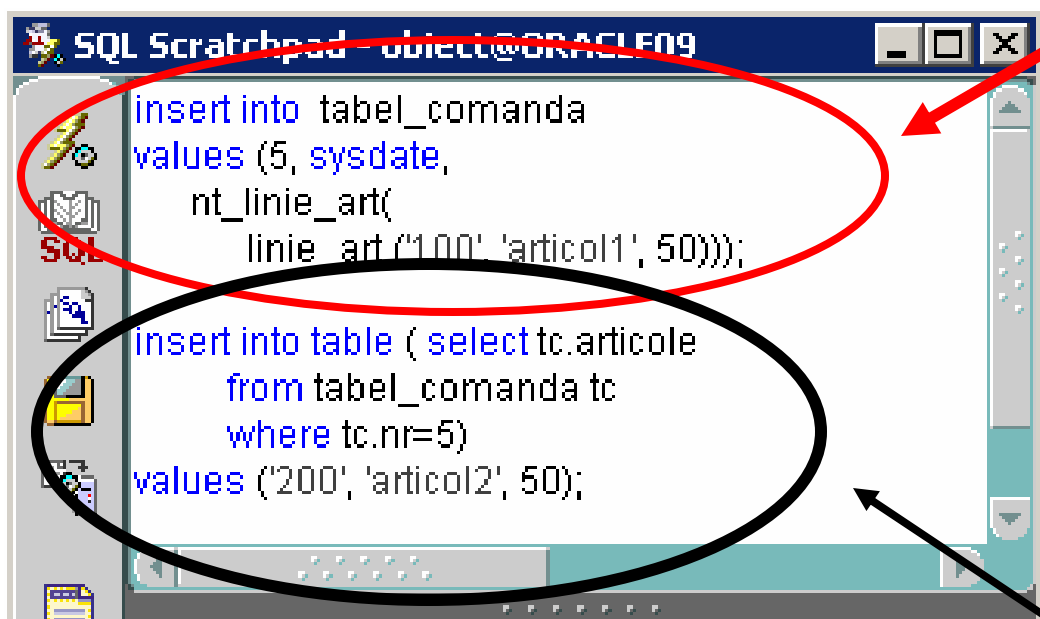
PLANETS(NAME, MASS, SATELLITES(NAME, DIAMETER))	

Sun	23
NT_PL_T(PLANET_T('Neptune', 10, NT_SAT_T(SATELLITE_T('Proteus', 67), SATELLITE_T('Triton', 82))), PLANET_T('Jupiter', 189, NT_SAT_T(SATELLITE_T('Callisto', 97), SATELLITE_T('Ganymede', 22))))	

Executarea operațiilor DML asupra datelor de tip colecție

- Oracle suportă următoarele tipuri de operații asupra coloanelor care conțin tabele încapsulate:
 - Inserări și actualizări care furnizează o valoare nouă pentru întreaga colecție
 - Actualizări pe componente
 - Inserarea de noi elemente în colecție
 - Ștergerea de elemente din colecție
 - Actualizarea elementelor colecției
 - Pentru actualizarea coloanelor tabelelor încapsulate instrucțiunile DML identifică tabelul încapsulat asupra căruia se operează prin utilizarea expresiei TABLE
- NU SUPORTĂ actualizări parțiale pe coloane de tip varray

Exemplu – operații parțiale asupra coloanelor de tip tabel încapsulat - INSERARE



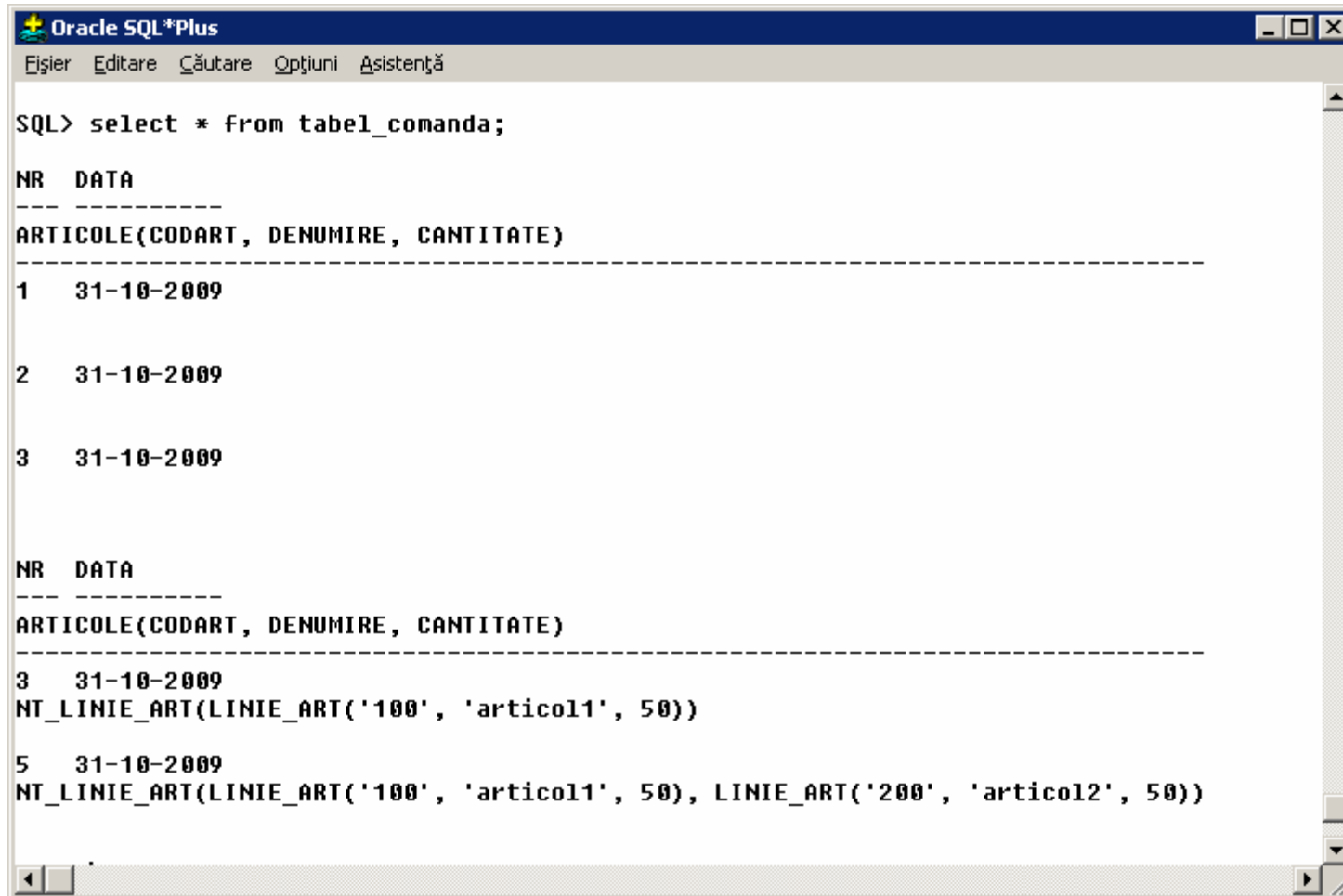
```
SQL Scratchpad - Object@ORACLE09
insert into tabel_comanda
values (5, sysdate,
       nt_linie_art(
         linie_art('100' 'articol1', 50)));

insert into table ( select tc.articole
                   from tabel_comanda tc
                   where tc.nr=5)
values ('200', 'articol2', 50);
```

s-au inserat date in tabelul
“tabel_comanda” care
conține un tabel
încapsulat de tip linie_art

S-a făcut o inserare
in tabelul încapsulat
aferent rândului cu
nr=5

Rezultatul :



The image shows a screenshot of the Oracle SQL*Plus command-line interface. The window title is 'Oracle SQL*Plus'. The menu bar includes 'Fișier', 'Editare', 'Căutare', 'Opțiuni', and 'Asistență'. The command prompt shows the query 'SQL> select * from tabel_comanda;'. The output is displayed in two sections, each starting with a header 'NR DATA' followed by a dashed line and the column list 'ARTICOLE(CODART, DENUMIRE, CANTITATE)'. The first section contains three rows of data, and the second section contains two rows of data, including nested function calls.

```
Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență

SQL> select * from tabel_comanda;

NR DATA
-----
ARTICOLE(CODART, DENUMIRE, CANTITATE)
-----
1 31-10-2009

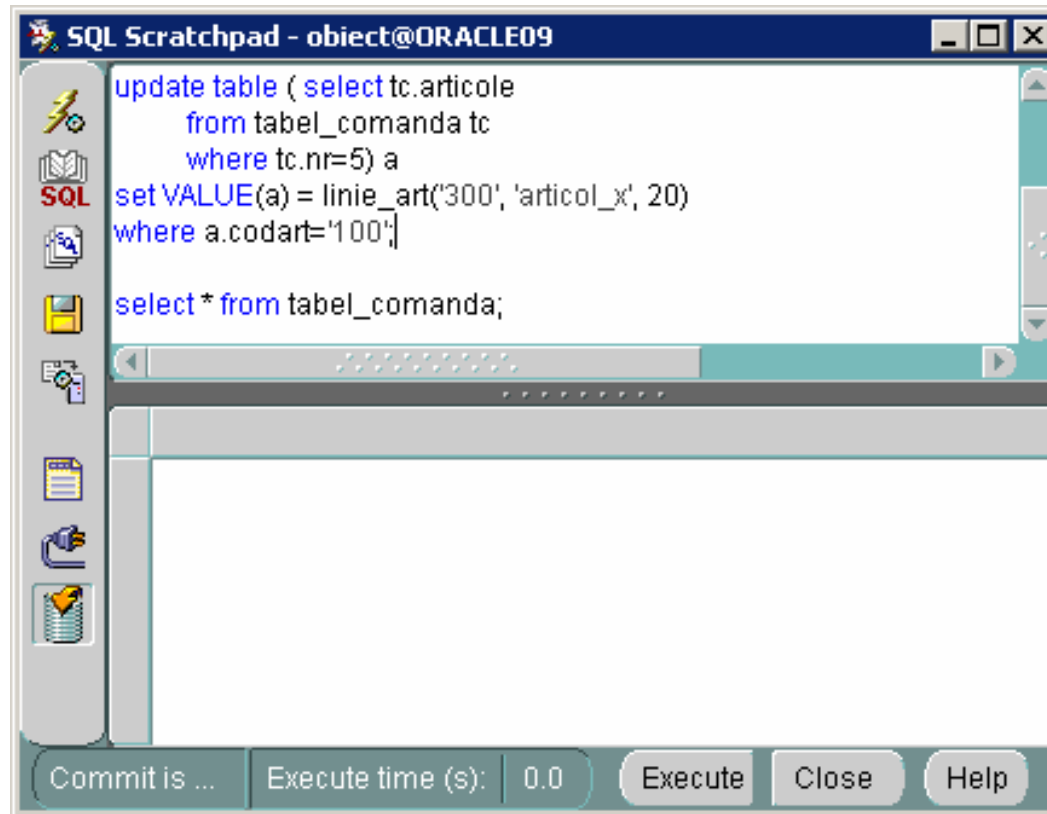
2 31-10-2009

3 31-10-2009

NR DATA
-----
ARTICOLE(CODART, DENUMIRE, CANTITATE)
-----
3 31-10-2009
NT_LINIE_ART(LINIE_ART('100', 'articol1', 50))

5 31-10-2009
NT_LINIE_ART(LINIE_ART('100', 'articol1', 50), LINIE_ART('200', 'articol2', 50))
```

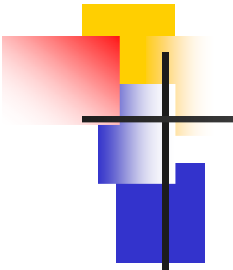
Exemplu – operații parțiale asupra coloanelor de tip tabel încapsulat - ACTUALIZARE



The screenshot shows a window titled "SQL Scratchpad - obiect@ORACLE09". The window contains an SQL editor with the following text:

```
update table ( select tc.articole  
                from tabel_comanda tc  
                where tc.nr=5) a  
set VALUE(a) = linie_art('300', 'articol_x', 20)  
where a.codart='100';  
  
select * from tabel_comanda;
```

The editor has a toolbar on the left with icons for SQL, file operations, and execution. At the bottom, there is a status bar with the text "Commit is ..." and "Execute time (s): 0.0", along with buttons for "Execute", "Close", and "Help".



```
Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență
SQL> select * from tabel_comanda;

NR  DATA
-----
ARTICOLE(CODART, DENUMIRE, CANTITATE)
-----
1   31-10-2009

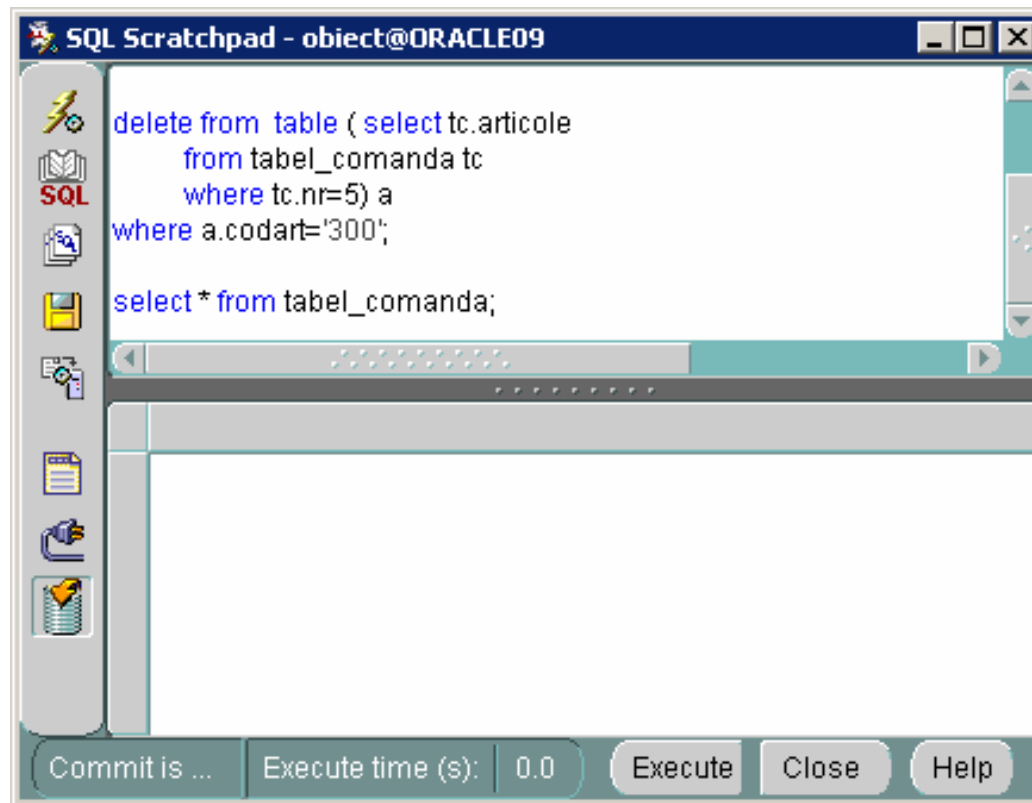
2   31-10-2009

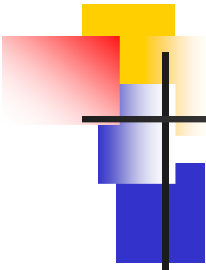
3   31-10-2009

NR  DATA
-----
ARTICOLE(CODART, DENUMIRE, CANTITATE)
-----
3   31-10-2009
NT_LINIE_ART(LINIE_ART('100', 'articol1', 50))

5   31-10-2009
NT_LINIE_ART(LINIE_ART('300', 'articol_x', 20), LINIE_ART('200', 'articol2', 50)
)
```

Exemplu – operații parțiale asupra coloanelor de tip tabel încapsulat - ȘTERGERE





```
Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență
SQL> select * from tabel_comanda;

NR  DATA
-----
ARTICOLE(CODART, DENUMIRE, CANTITATE)
-----
1   31-10-2009

2   31-10-2009

3   31-10-2009

NR  DATA
-----
ARTICOLE(CODART, DENUMIRE, CANTITATE)
-----
3   31-10-2009
NT_LINIE_ART(LINIE_ART('100', 'articol1', 50))

5   31-10-2009
NT_LINIE_ART(LINIE_ART('200', 'articol2', 50))

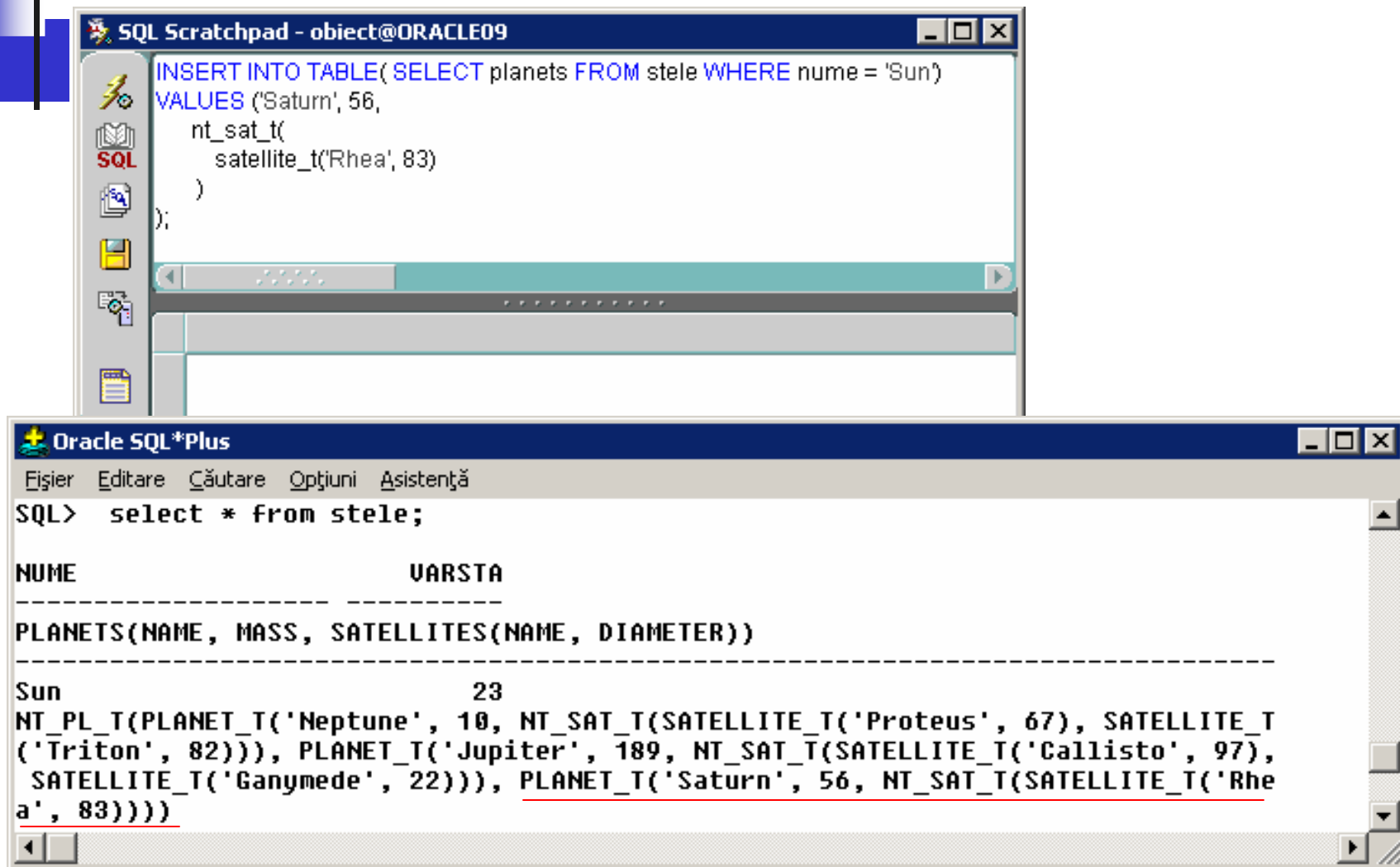
SQL> |
```



Operații DML asupra colecțiilor multinivel

- Pentru colecțiile multinivel de tabele încapsulate operațiile pot fi realizate atomic, asupra colecției văzute ca un întreg sau pe bucăți asupra elementelor selectate
- Pentru colecțiile multinivel de tip varray operațiile DML pot fi executate numai atomic.

Exemplu 1



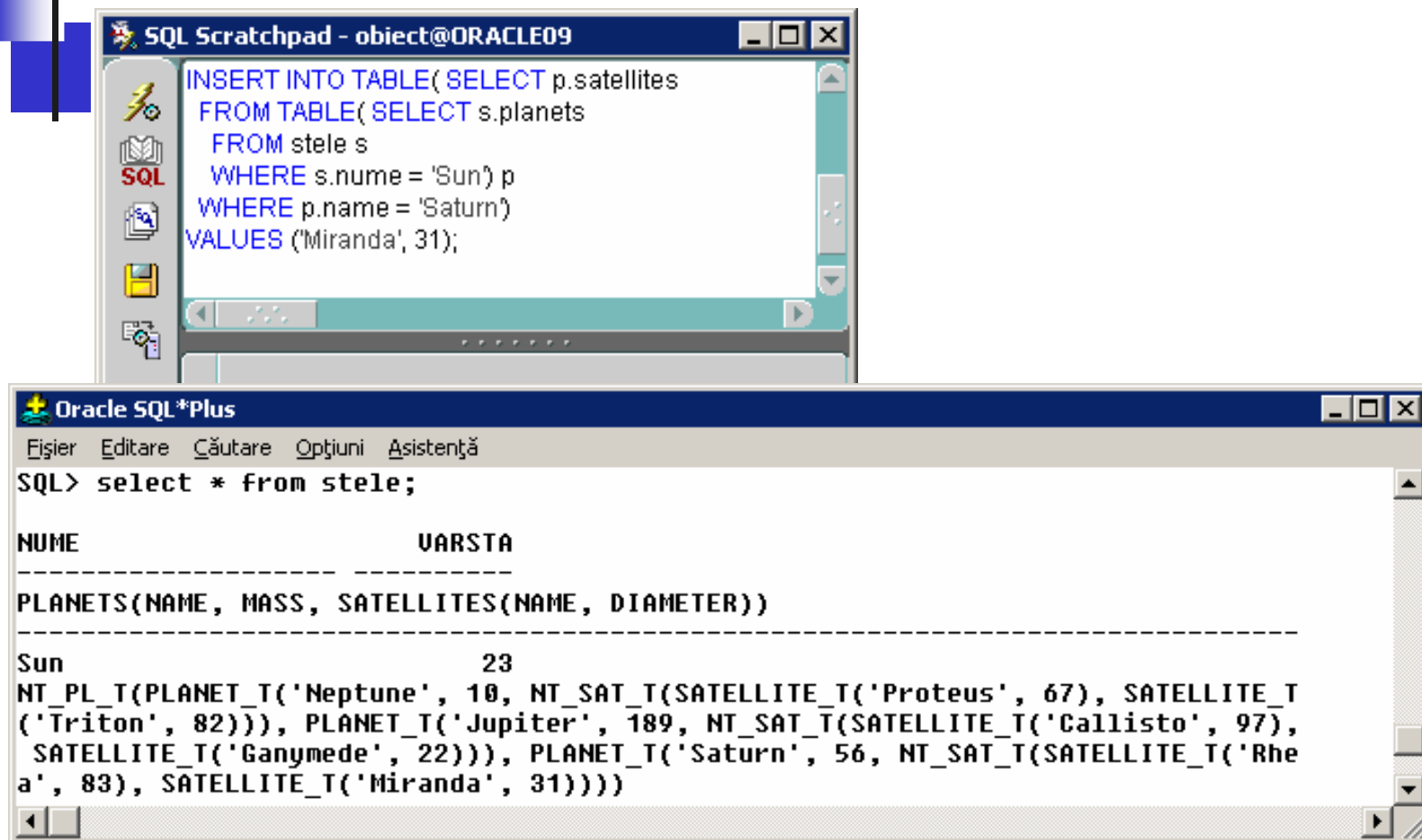
The image shows two overlapping windows from an Oracle database environment. The top window, titled "SQL Scratchpad - obiect@ORACLE09", contains an SQL statement: `INSERT INTO TABLE(SELECT planets FROM stele WHERE nume = 'Sun') VALUES ('Saturn', 56, nt_sat_t(satellite_t('Rhea', 83)));`. The bottom window, titled "Oracle SQL*Plus", shows the execution of `SQL> select * from stele;`. The output displays the structure of the 'stele' table, including nested tables for planets and satellites, and lists the data for the 'Sun' entry, including its satellites 'Proteus' and 'Rhea'.

```
SQL Scratchpad - obiect@ORACLE09
INSERT INTO TABLE( SELECT planets FROM stele WHERE nume = 'Sun')
VALUES ('Saturn', 56,
        nt_sat_t(
            satellite_t('Rhea', 83)
        )
);

Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență
SQL> select * from stele;

NUME                                VARSTA
-----
PLANETS(NAME, MASS, SATELLITES(NAME, DIAMETER))
-----
Sun                                23
NT_PL_T(PLANET_T('Neptune', 10, NT_SAT_T(SATELLITE_T('Proteus', 67), SATELLITE_T
('Triton', 82))), PLANET_T('Jupiter', 189, NT_SAT_T(SATELLITE_T('Callisto', 97),
SATELLITE_T('Ganymede', 22))), PLANET_T('Saturn', 56, NT_SAT_T(SATELLITE_T('Rhe
a', 83))))
```


Exemplu 2- inserare în tabelul interior



The image shows two overlapping windows from an Oracle database environment. The top window, titled "SQL Scratchpad - obiect@ORACLE09", contains an SQL statement for inserting data into a table. The bottom window, titled "Oracle SQL*Plus", shows the execution of a query that displays the structure and data of a table named "stele".

SQL Scratchpad - obiect@ORACLE09

```
INSERT INTO TABLE( SELECT p.satellites
FROM TABLE( SELECT s.planets
FROM stele s
WHERE s.num = 'Sun') p
WHERE p.name = 'Saturn')
VALUES ('Miranda', 31);
```

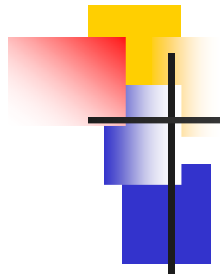
Oracle SQL*Plus

Fișier Editare Căutare Opțiuni Asistență

```
SQL> select * from stele;
```

NUME	VARSTA
Sun	23
PLANETS(NAME, MASS, SATELLITES(NAME, DIAMETER))	

NT_PL_T(PLANET_T('Neptune', 10, NT_SAT_T(SATELLITE_T('Proteus', 67), SATELLITE_T('Triton', 82))), PLANET_T('Jupiter', 189, NT_SAT_T(SATELLITE_T('Callisto', 97), SATELLITE_T('Ganymede', 22))), PLANET_T('Saturn', 56, NT_SAT_T(SATELLITE_T('Rhea', 83), SATELLITE_T('Miranda', 31))))	



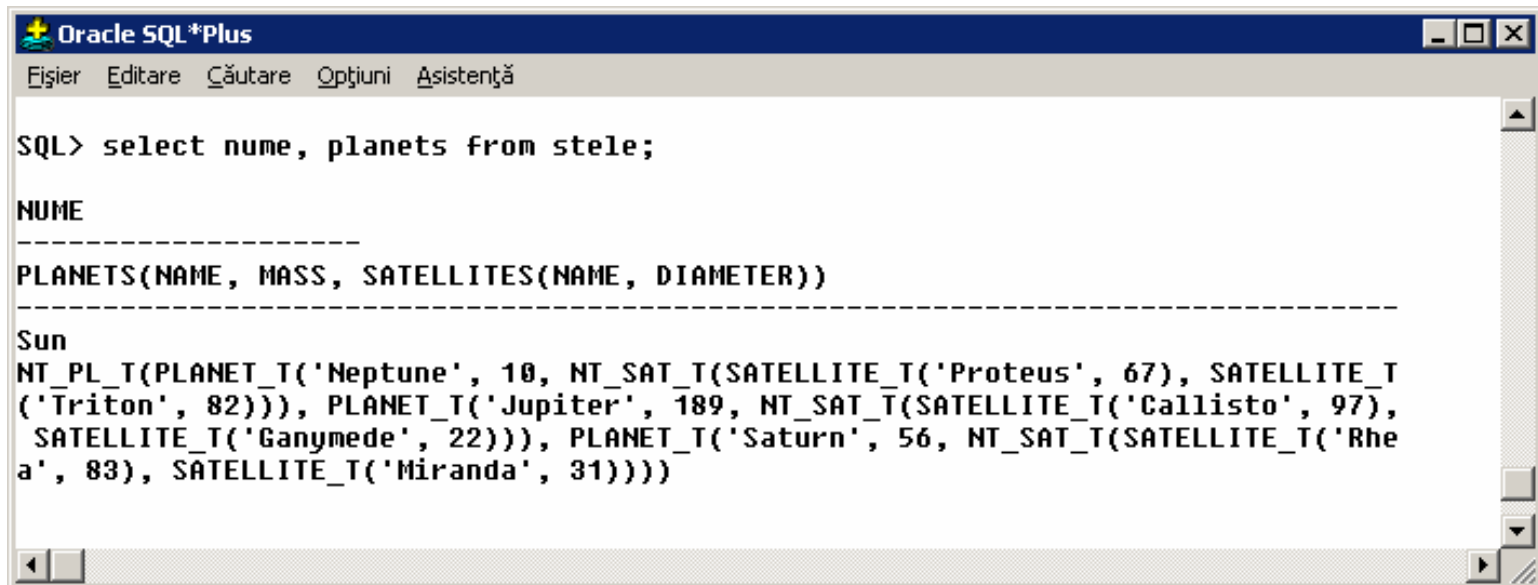
Interogarea colecțiilor

- Există două modalități de a interoga un tabel care conține un atribut sau o coloană de tip colecție
 - Colecția poate fi returnată înglobat în rândul rezultat care o conține (*nested collection*)
 - Colecția este distribuită (*unnested*) astfel încât fiecare element apare separat în rândul rezultat

Interogări care întorc rezultatul înglobat

1. Select * from stele;

2.



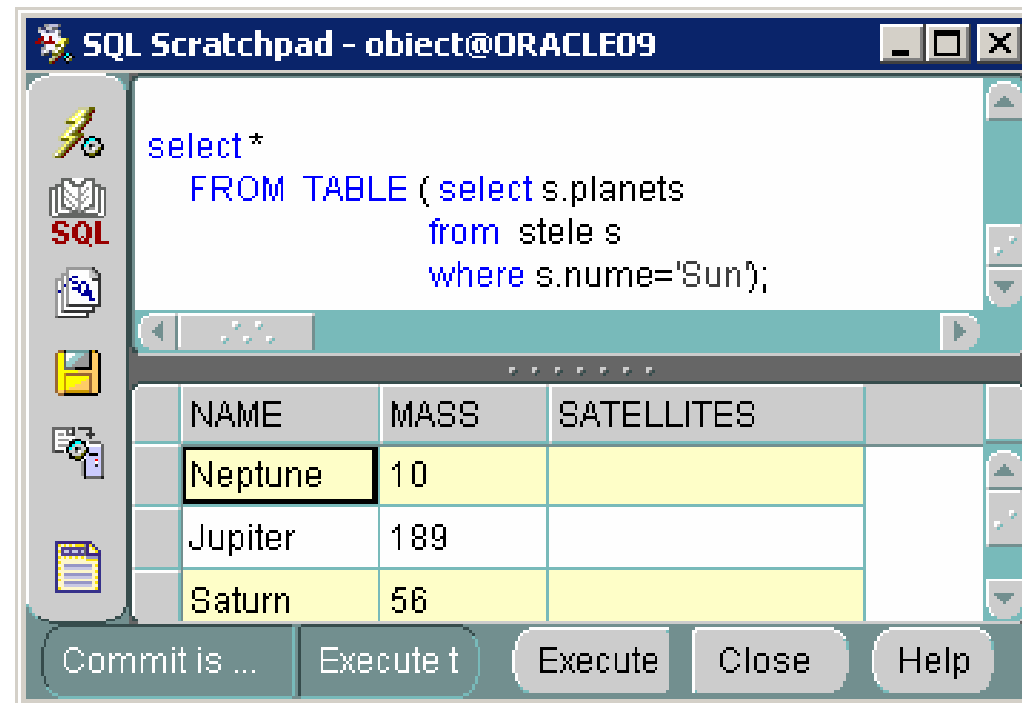
The screenshot shows the Oracle SQL*Plus interface. The command prompt shows the query: `SQL> select nume, planets from stele;`. The output displays the column `NUME` followed by a separator line, then the column `PLANETS(NAME, MASS, SATELLITES(NAME, DIAMETER))` followed by another separator line. The first row of data is for the 'Sun', with the `PLANETS` column containing a nested SQL query: `NT_PL_T(PLANET_T('Neptune', 10, NT_SAT_T(SATELLITE_T('Proteus', 67), SATELLITE_T('Triton', 82))), PLANET_T('Jupiter', 189, NT_SAT_T(SATELLITE_T('Callisto', 97), SATELLITE_T('Ganymede', 22))), PLANET_T('Saturn', 56, NT_SAT_T(SATELLITE_T('Rhea', 83), SATELLITE_T('Miranda', 31))))`.

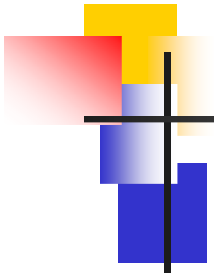
```
Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență

SQL> select nume, planets from stele;

NUME
-----
PLANETS(NAME, MASS, SATELLITES(NAME, DIAMETER))
-----
Sun
NT_PL_T(PLANET_T('Neptune', 10, NT_SAT_T(SATELLITE_T('Proteus', 67), SATELLITE_T
('Triton', 82))), PLANET_T('Jupiter', 189, NT_SAT_T(SATELLITE_T('Callisto', 97),
SATELLITE_T('Ganymede', 22))), PLANET_T('Saturn', 56, NT_SAT_T(SATELLITE_T('Rhe
a', 83), SATELLITE_T('Miranda', 31))))
```

Interogări care întorc rezultate detaliat pe elemente (unnested)





- Restricții referitoare la utilizarea subinterogărilor într-o expresie TABLE
 - Subinterogarea trebuie să întoarcă un tip colecție
 - Lista SELECT a subinterogării trebuie să conțină exact un articol
 - Subinterogarea trebuie să întoarcă numai o singură colecție (nu poate întoarce colecții pentru mai multe rânduri)

Interogarea colecțiilor multinivel

- Se folosesc doar interogări care întorc rezultate detaliate
- Se pot utiliza atât pentru vectori variabili cât și pentru tabele încapsulate

