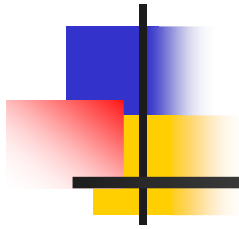


# CURS 4



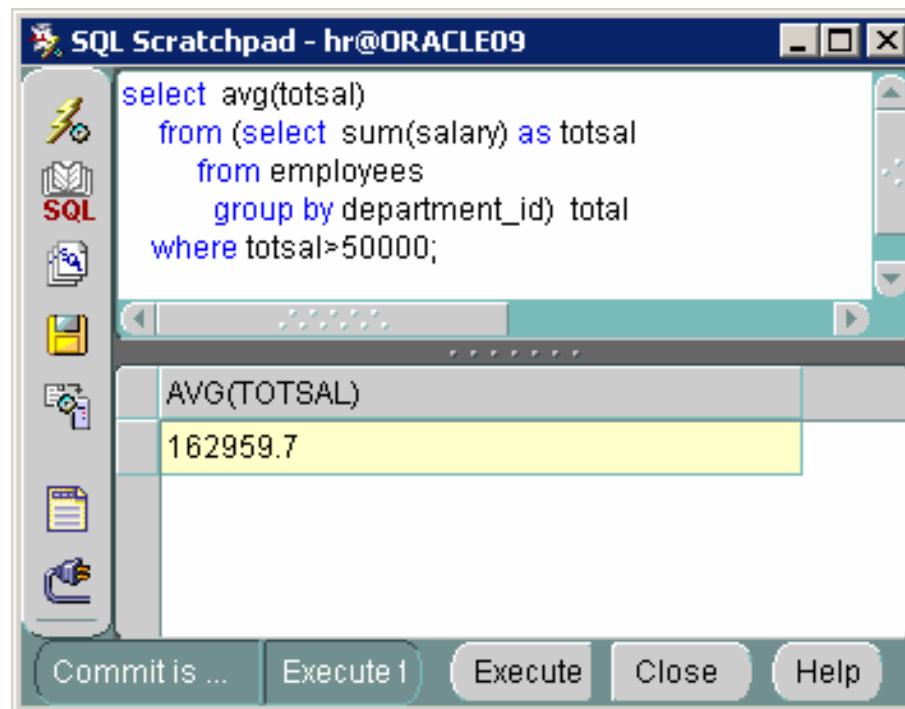
## SQL avansat I Componenta obiect – relațională în Oracle

# SQL avansat - tabele derivate

- SQL permite utilizarea unei subinterogări în clauza FROM a unei fraze SELECT (vedere *inline*)

Să se găsească media totalurilor salariilor pentru acele departamente unde aceste totaluri depășesc 50000.

**Obs:** nu este necesar să se folosească clauza HAVING, deoarece mai întâi se determină o relație temporară *total* în clauza FROM ale cărei valori din câmpul *totsal* care depășesc valoarea de 50000 vor fi luate în considerare pentru calculul mediei finale



The screenshot shows a window titled "SQL Scratchpad - hr@ORACLE09". The query entered is:

```
select avg(totsal)
from (select sum(salary) as totsall
from employees
group by department_id) total
where totsall>50000;
```

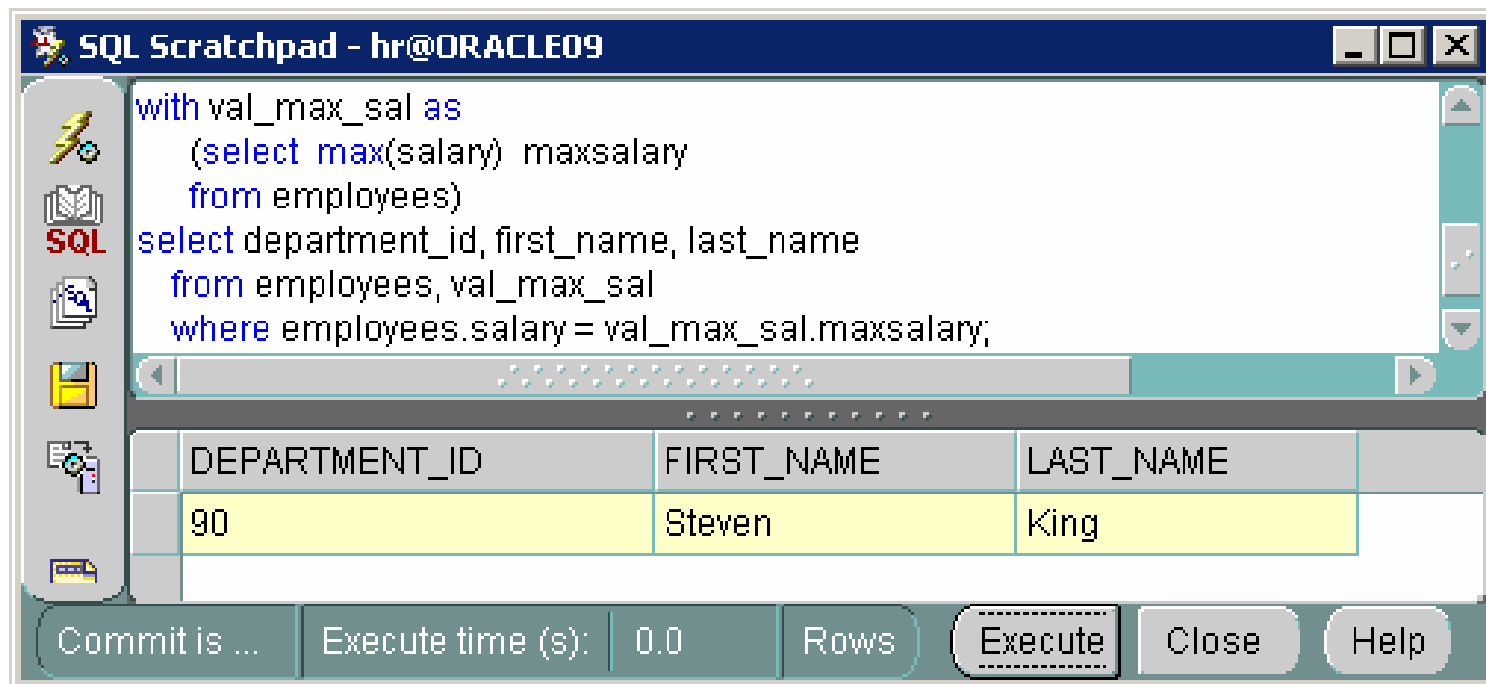
The result is displayed in a table with one row:

AVG(TOTSAL)
162959.7

At the bottom of the window, there are buttons for "Commit is ...", "Execute 1", "Execute", "Close", and "Help".

# SQL avansat - clauza WITH

- Furnizează o modalitate de a defini o vedere temporară a cărei definiție este disponibilă numai pentru interogarea în care apare clauza WITH
- Să se găsească numele si departamentele angajaților care au salariul egal cu maximul din companie.





---

# UDT



# Tipuri native de date in Oracle - SQL

VARCHAR2(n)	RAW(n)
NUMBER(n,d)	ROWID
DATE	CHAR
TIMESTAMP	CLOB
INTERVAL YEAR TO MONTH	BLOB
INTERVAL DAY TO SECOND	BFILE



# Tipuri definite de utilizatori (UDT - User Defined Types)

---

- SGBD-urile care conțin dialecte SQL avansate permit definirea de către utilizatori a propriilor tipuri de date
  - este o facilitare ce se dorește o extensie către modelul obiectual.
  - utilizează tipurile de date predefinite în sistem sau alte tipuri de date definite de utilizatori ca și elemente cu care se construiesc noi tipuri pentru a modela structura și comportamentul datelor în aplicații
- sunt de mai multe categorii
  - tip obiect
  - referinte
  - colecții
    - Vectori
    - Tabele încapsulate

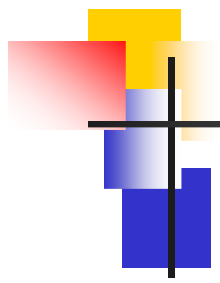


# UDT – tipul obiect

- reprezintă abstractizări ale entităților din lumea reală
- este un obiect schemă cu trei elemente componente:
  - *numele* care identifica în mod unic tipul respectiv în schema
  - *n attribute* , care sunt de tip predefinit în sistem sau sunt alte tipuri definite de utilizator. Caracterizează și modelează structura entității reale
  - *n metode* - sunt funcții sau proceduri scrise în PL/SQL și memorate în baza de date , sau scrise în limbaje precum C sau Java și memorate extern bazei de date
    - implementează operațiile pe care o aplicație le poate executa asupra entității modelate
- poate fi utilizat în același mod ca și tipurile de date predefinite cum ar fi NUMBER sau VARCHAR2
- o valoare a unui tip obiect este o instanță a acestui tip
- instanța unui tip obiect este denumită *obiect*.

Instrucțiunea de creare are sintaxa:

**CREATE TYPE *nume\_tip* AS [OBJECT] *definiție***



# Moștenirea tipului

- Un tip obiect poate fi specializat prin crearea de **subtipuri** derivate dintr-un tip obiect părinte numit **supertip**
  - Acestea au attribute sau metode suplimentare ca și caracteristici de diferențiere
- Subtipurile și supertipurile sunt legate prin **moștenire**
  - Ca versiuni specializate ale părinților, subipurile au toate attributele și metodele părinților plus alte specificații definite în subtip.
  - Subtipurile și supertipurile, conectate prin moștenire creează o ierarhie a tipurilor.



# Object tabel

- Un *object tabel* este un tip special de tabel în care fiecare rând reprezintă un obiect
- Exemplu

The image displays two Oracle database management interfaces. The top window, titled "SQL Scratchpad - hr@ORACLE09", shows the following SQL command:

```
CREATE TYPE person AS OBJECT (  
  name   VARCHAR2(30),  
  phone  VARCHAR2(20) );
```

The bottom window, titled "Oracle Enterprise Manager Console, Standalone", shows the configuration of the "PERSON" object type. The left pane shows the "User Types" hierarchy with "PERSON" selected under "Object Types". The right pane shows the "General" tab with the following details:

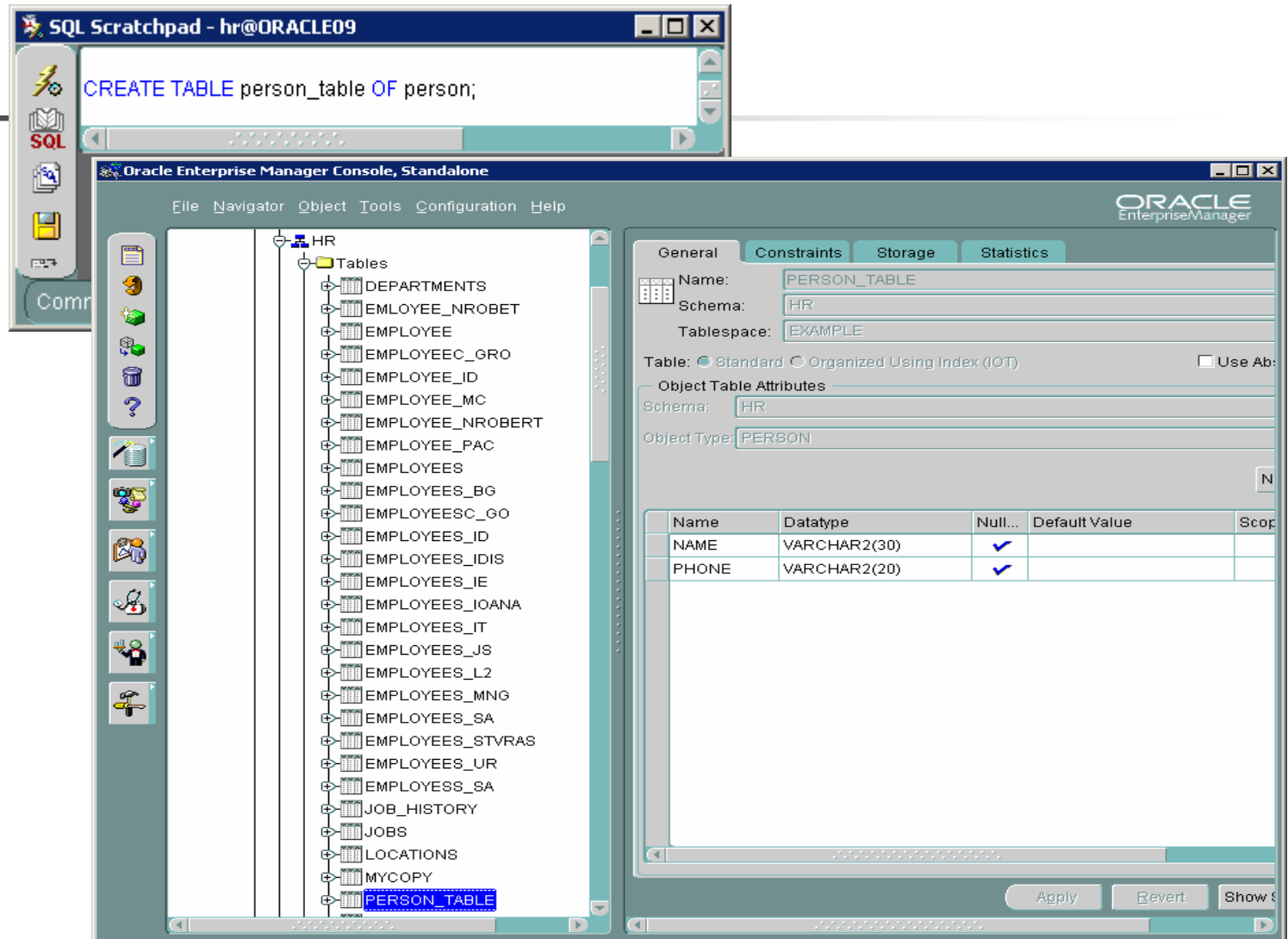
- Name: PERSON
- Schema: HR
- Show Errors button
- Attributes table:

Name	Datatype
NAME	VARCHAR2(30)
PHONE	VARCHAR2(20)

- Buttons: Add, Remove, Edit
- Methods table:

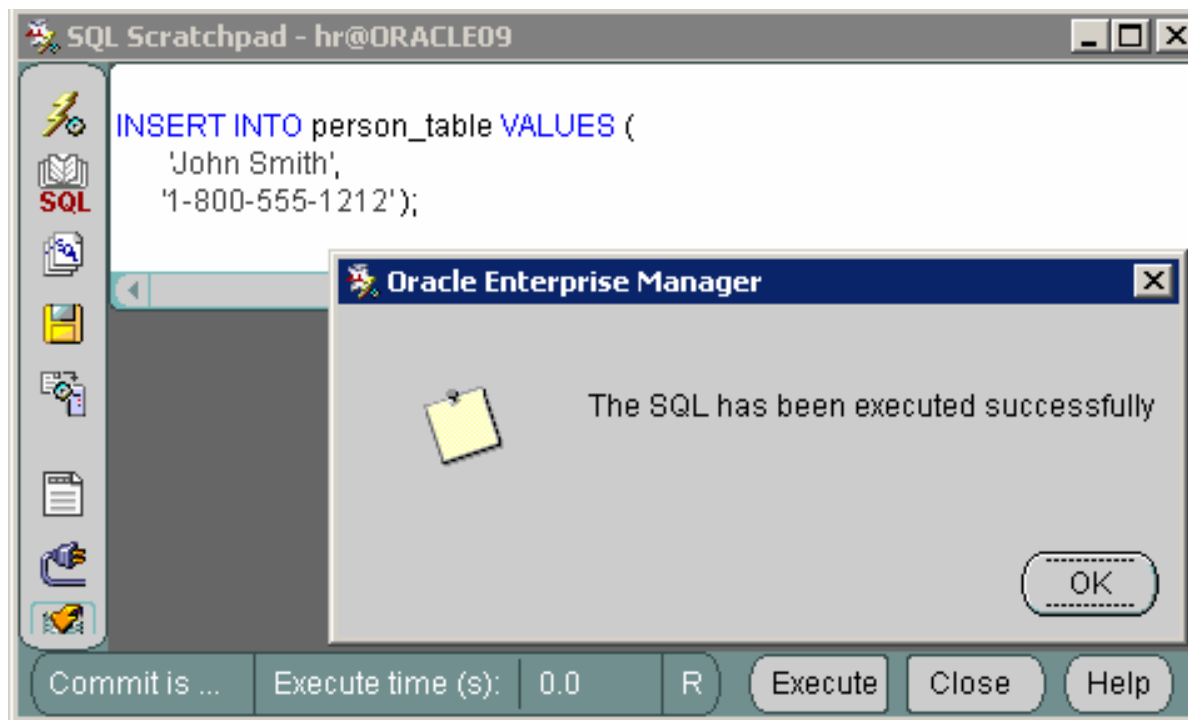
Name	Method Type
------	-------------

- Buttons: Add, Remove

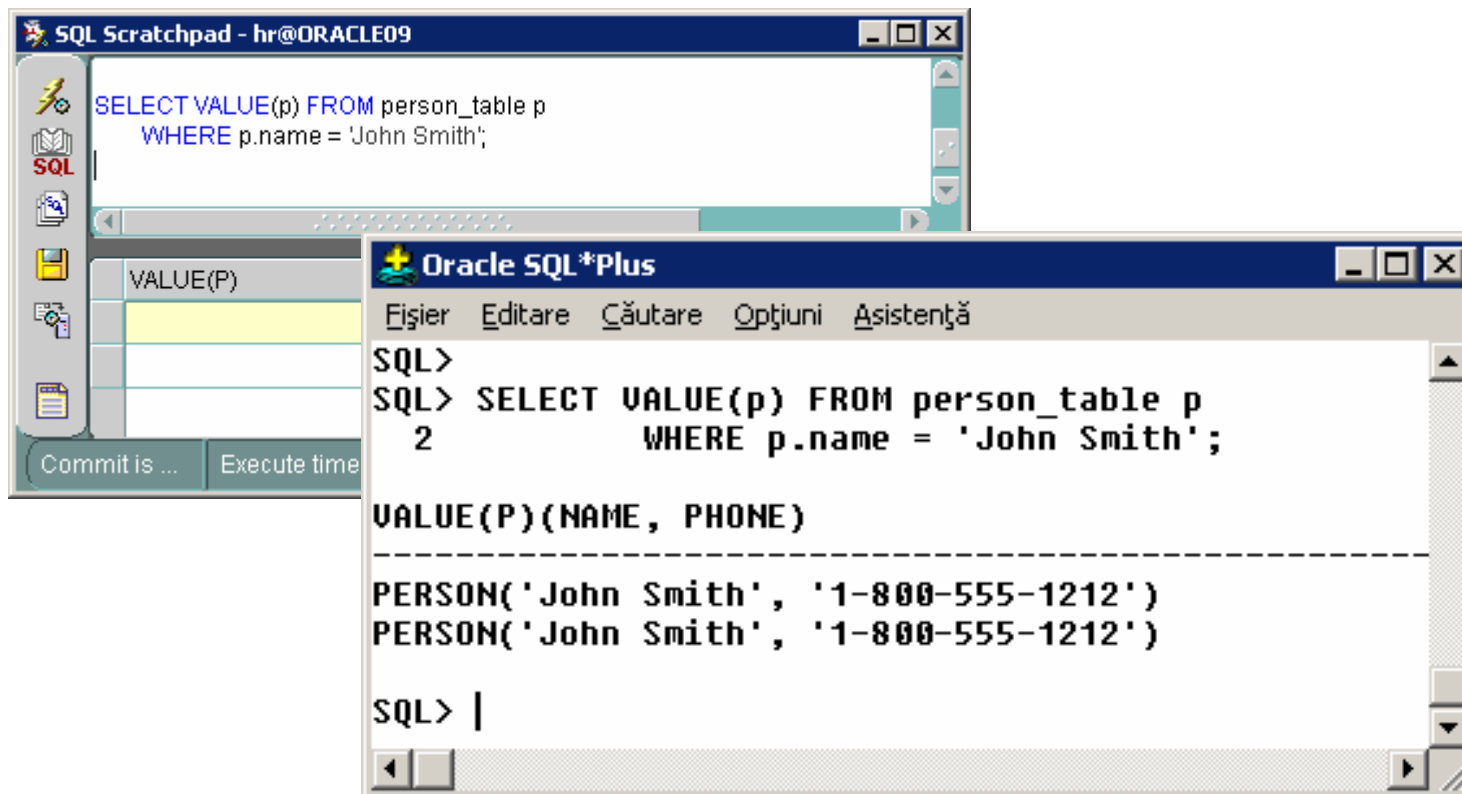


- Tabelul creat poate fi văzut în două moduri:

1. Ca un tabel multi-coloana în care fiecare atribut al tipului de obiect person (name si phone), ocupă o coloană si este permisă execuția operațiilor relaționale



2. Ca un tabel cu o singura coloana în care fiecare rând este un obiect *person*, asupra căruia sunt permise operații orientate obiect



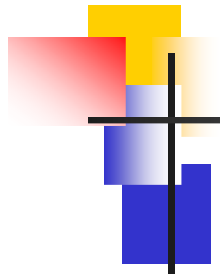
The image shows two overlapping windows from an Oracle database environment. The top window, titled "SQL Scratchpad - hr@ORACLE09", contains the SQL query: `SELECT VALUE(p) FROM person_table p WHERE p.name = 'John Smith';`. The bottom window, titled "Oracle SQL\*Plus", shows the execution of this query. It displays the prompt `SQL>`, the query text, and the results. The results are shown in a table with one column, `VALUE(P)`, and two rows of data: `PERSON('John Smith', '1-800-555-1212')` and `PERSON('John Smith', '1-800-555-1212')`. The window also shows a menu bar with options like "Fișier", "Editare", "Căutare", "Opțiuni", and "Asistență".

```
SQL Scratchpad - hr@ORACLE09
SELECT VALUE(p) FROM person_table p
WHERE p.name = 'John Smith';

Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență
SQL>
SQL> SELECT VALUE(p) FROM person_table p
      2          WHERE p.name = 'John Smith';

VALUE(P)(NAME, PHONE)
-----
PERSON('John Smith', '1-800-555-1212')
PERSON('John Smith', '1-800-555-1212')

SQL> |
```

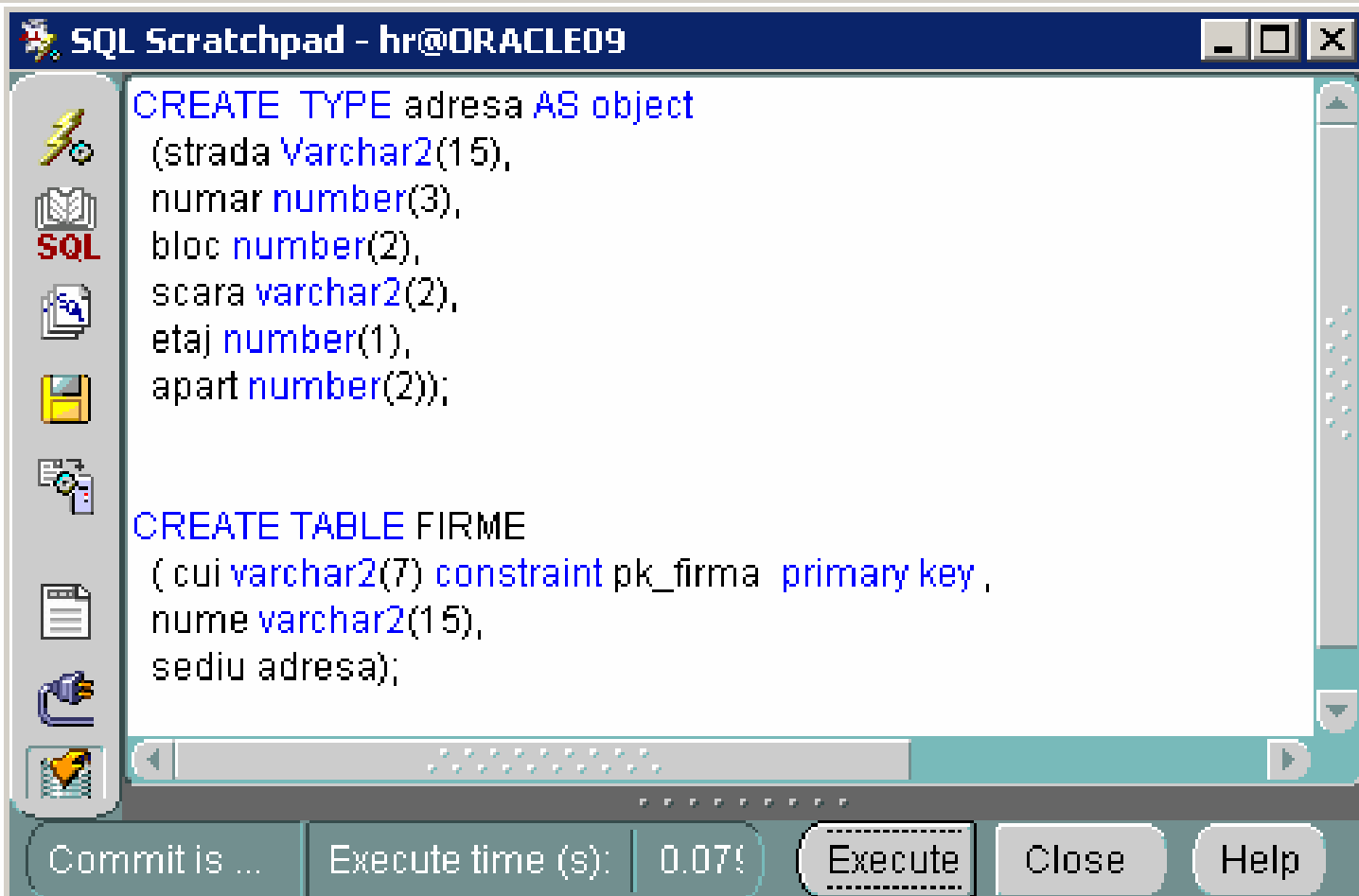


# Obiecte rând și obiecte coloană

---

- Obiectele care ocupă un rând complet in tabele obiect se numesc *obiecte rând*.
- Obiectele care ocupă coloane in tabele cu rânduri mai complexe, sau sunt attribute ale altor obiecte se numesc *obiecte coloană*.

## Exemplu – obiect coloana

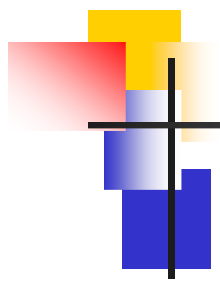


The screenshot shows a window titled "SQL Scratchpad - hr@ORACLE09". The window contains two SQL statements. The first statement creates a type named "adresa" with columns: "strada" (Varchar2(15)), "numar" (number(3)), "bloc" (number(2)), "scara" (varchar2(2)), "etaj" (number(1)), and "apart" (number(2)). The second statement creates a table named "FIRME" with columns: "cui" (varchar2(7) primary key), "nume" (varchar2(15)), and "sediu" (adresa). 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 (s): 0.07!", "Execute", "Close", and "Help".

```
CREATE TYPE adresa AS object
(strada Varchar2(15),
numar number(3),
bloc number(2),
scara varchar2(2),
etaj number(1),
apart number(2));

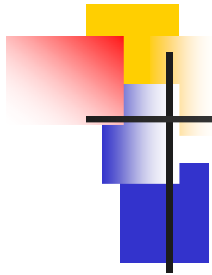
CREATE TABLE FIRME
(cui varchar2(7) constraint pk_firma primary key ,
nume varchar2(15),
sediu adresa);
```

Commit is ... Execute time (s): 0.07! Execute Close Help



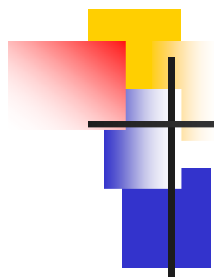
# Obiect vedere

- Un obiect vedere reprezintă un mod de a accesa date relaționale utilizând caracteristici obiect-relaționale
  - In acest mod este posibilă dezvoltarea de aplicații orientate obiect fără a schimba schema relațională de bază.
- Așa cum o vedere relațională este un tabel relațional virtual, un obiect vedere este un obiect tabel virtual.
  - fiecare rând din vedere este un obiect



- Obiectele vedere sunt utile pentru realizarea de aplicații orientate obiect deoarece datele din vedere pot fi preluate din tabele relaționale și accesate ca și cum tabelul ar fi fost definit ca un obiect tabel.
  - Se pot rula aplicații orientate obiect fără a converti tabelele existente într-o structură fizică diferită.
- Pot fi utilizate, ca și vederile relaționale pentru a prezenta selectiv datele
- Pot conduce către performanțe mai bune
  - Ex. datele relaționale care constituie un rând într-un obiect vedere parcurg rețeaua ca o unitate evitându-se astfel transferuri multiple.



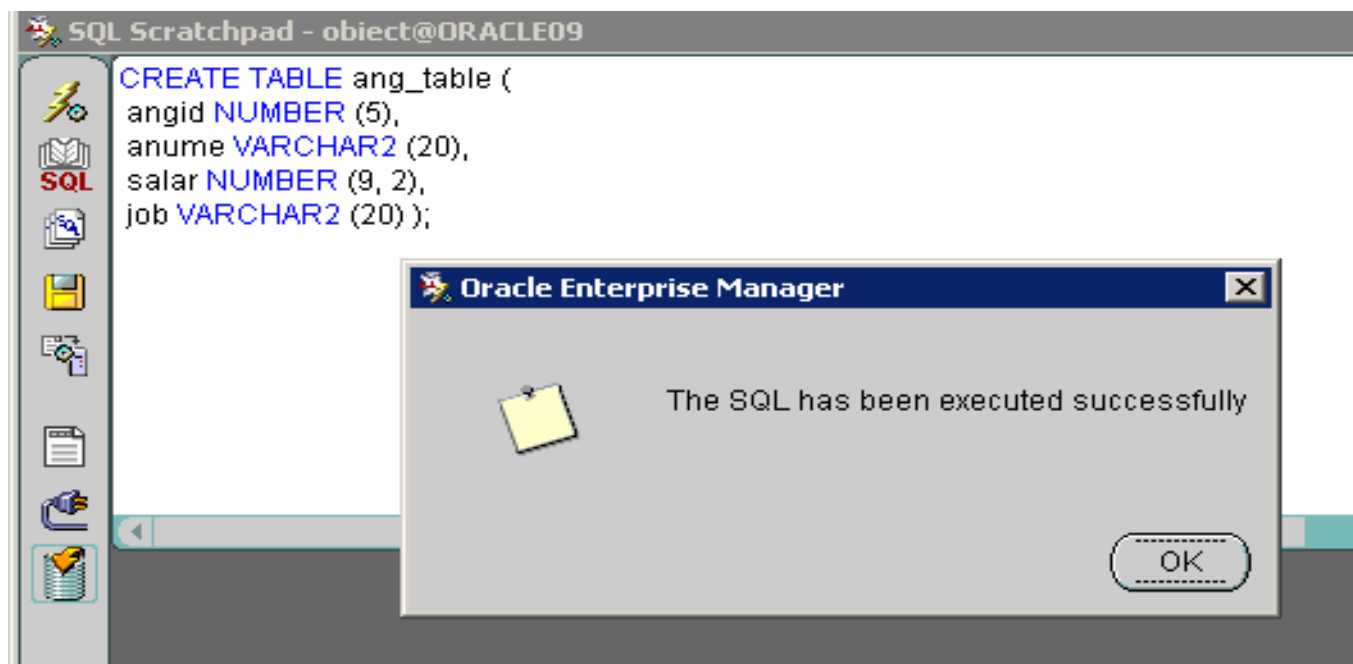


# Definirea unui obiect vedere

- Procesul de definire a unui obiect vedere cuprinde următorii pași
  1. Se definește un **tip obiect** în care fiecare atribut al tipului corespunde unei coloane existente dintr-un tabel relațional.
  2. Se scrie o **interogare** prin care se specifică modul de extragere a datelor din tabelul relațional.
    1. Coloanele trebuie specificate în aceeași ordine în care apar ca attribute în tipul obiect.
  3. Se specifică o **valoare unică** bazată pe attribute ale tabelelor de bază care să fie identificatorul obiectului.
    1. Acestea permit crearea de pointeri (REF) către obiectele din vedere, Se poate utiliza în acest scop o cheie primară.
- Dacă se dorește ca obiectul vedere să fie actualizabil (???? CE PRESUPUNE ASTA???) iar attributele tipului obiect nu corespund exact coloanelor din tabelul de baza mai trebuie executată următoarea operație:
  1. Se scrie un trigger de tip INSTEAD OF care va fi executat oridecâte ori o aplicație încearcă să actualizeze datele in obiectul vedere.
- După creare vederea poate fi utilizată exact ca un obiect tabel

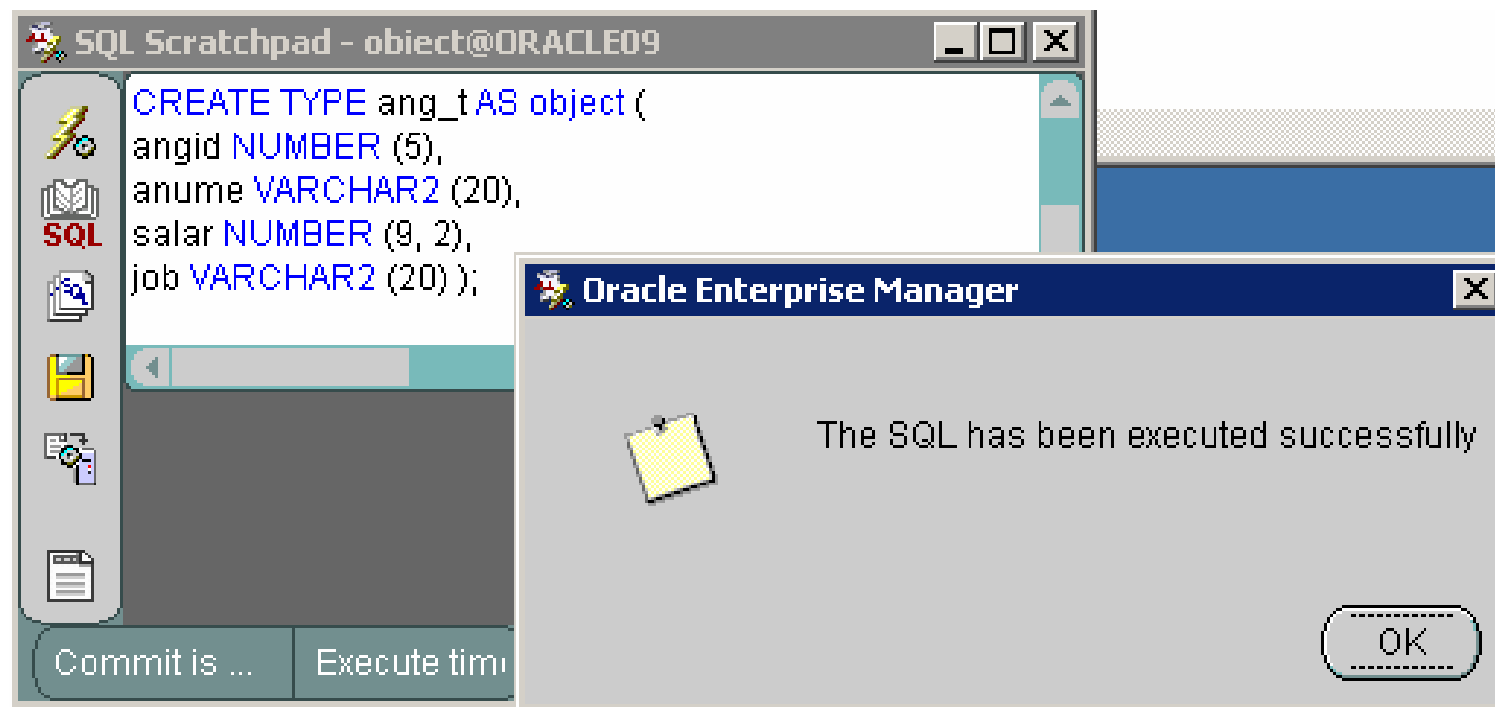
# Exemplu

- Se pornește de la tabelul *ang\_table* definit astfel:



# Exemplu

- Se definește tipul de data obiect *ang\_t*:



# Exemplu

- Se definește un obiect vedere în care fiecare rând al vederii este un obiect de tip *ang\_t*:
- Pentru a accesa data din coloana *angid* a tabelului relațional ar trebuie accesat atributul *angid* al tipului obiect.

The image displays two Oracle database management interfaces. The top window, titled "SQL Scratchpad - object@ORACLE09", shows the SQL command to create a view:

```
CREATE VIEW ang_view1 OF ang_t
WITH OBJECT IDENTIFIER (angid) AS
SELECT a.angid, a.anume, a.salar, a.job
FROM ang_table a
WHERE job = 'Developer';
```

The bottom window is Oracle Enterprise Manager. The left-hand "Network" tree shows the hierarchy: Databases > OEMREP > ORACLE09 - object > Schema > OBJECT > Views > ANG\_VIEW1. The right-hand pane shows the "Advanced" tab for the selected view, with the following details:

- Name: ANG\_VIEW1
- Schema: OBJECT
- Query Text:

```
SELECT a.angid, a.anume, a.salar, a.job FROM ang_table a
WHERE job = 'Developer'
```



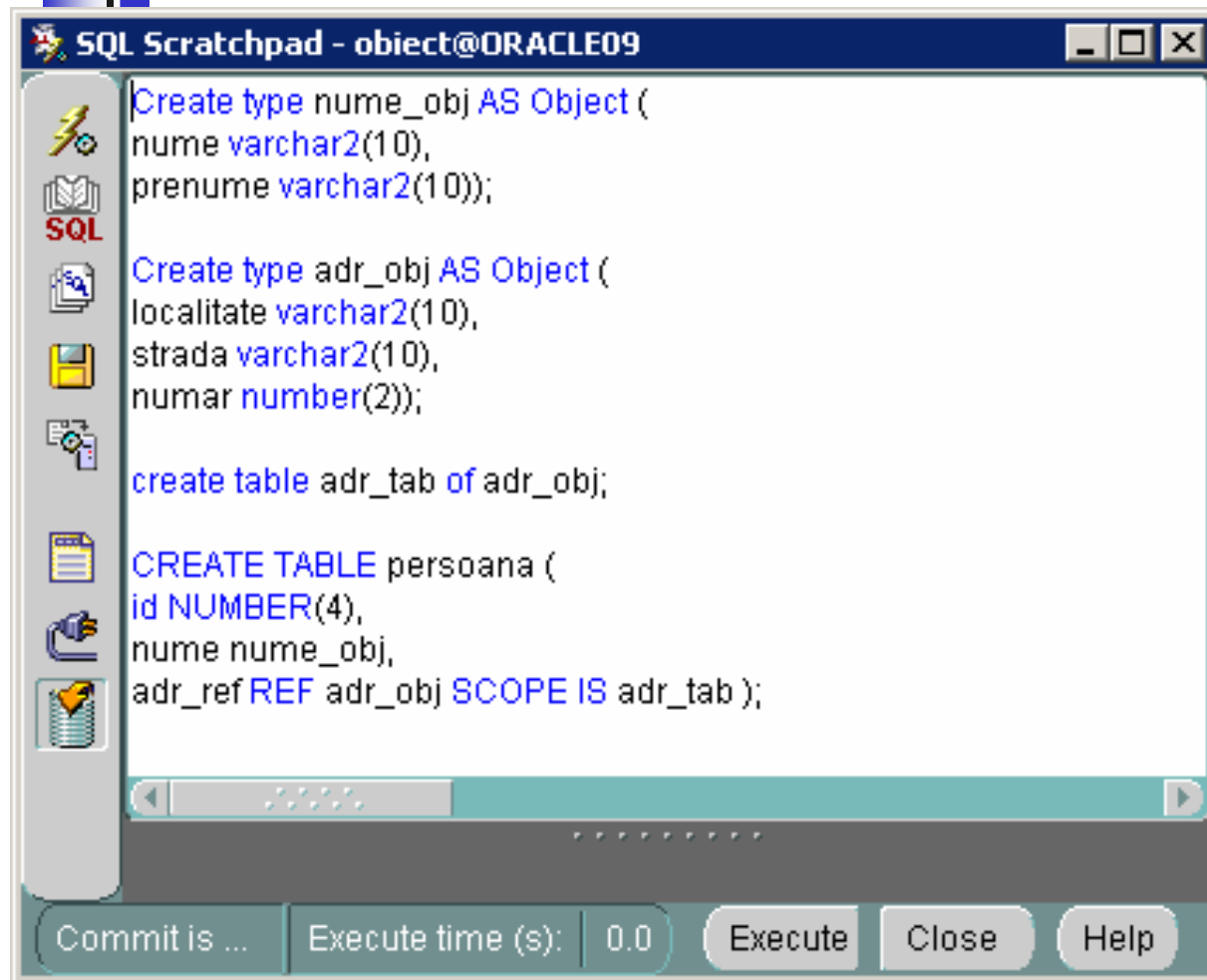

# Tip de data – REF (erințe)

- REF este un "pointer" logic către un obiect rând
- Datele de tip REF și colecțiile de date REF modelează asocierile dintre obiecte - în particular legăturile de tip 1:N -
  - Este redusă nevoia de definire a cheilor străine
- REF-urile furnizează un mecanism simplu pentru a naviga între obiecte.
  - Se poate folosi notația '*dot*' pentru a urmări pointerii.
  - La nevoie Oracle realizează joncțiunile iar în unele cazuri se evită efectuarea de joncțiuni.
- REF poate fi utilizat pentru:
  - a examina sau a actualiza obiectul referit
  - a obține o copie a obiectului referit
- Se poate schimba un REF astfel încât să indice un obiect diferit de același tip sau să îi atribuie o valoare *null*.



## Referințe limitate la un domeniu - (Scoped REF)

- La declararea unui element colecție sau a unui atribut al unui tip obiect ca REF există posibilitatea ca acesta să fie constrâns să conțină doar referințe la un obiect tabel specificat.
  - Acestea se numesc referințe limitate – *scoped REF*
  - Solicită mai puțin spațiu de stocare și permit un acces mai rapid decât tipul de referințe nelimitate



```
SQL Scratchpad - obiect@ORACLE09

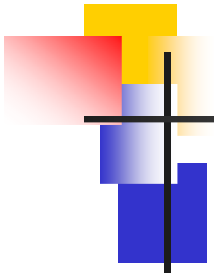
Create type nume_obj AS Object (
nume varchar2(10),
prenume varchar2(10));

Create type adr_obj AS Object (
localitate varchar2(10),
strada varchar2(10),
numar number(2));

create table adr_tab of adr_obj;

CREATE TABLE persoana (
id NUMBER(4),
nume nume_obj,
adr_ref REF adr_obj SCOPE IS adr_tab );
```

1. Se crează mai întâi tipurile de date (obiect) necesare (*nume\_obj* si *adr\_obj*)
2. Se crează obiectul tabel *adr\_tab*
3. Se creaza tabelul *persoana* care conține o coloana referință limitată (scoped) la tabelul *adr\_obj*



- un REF poate fi limitat la un obiect tabel de tipul declarat anterior (`adr_obj` în exemplu) sau la un obiect tabel construit pe orice subtip al tipului declarat.
  - In acest caz coloana REF este constrânsă să conțină doar referințe la instanțele subtipului (și ale subtipurilor sale dacă există) din tabel.



- Tabelul *persoana* creat are structura:

The screenshot displays the Oracle Enterprise Manager interface. On the left, the 'Navigator' pane shows the database structure: Databases > ORACLE09 - object > Instance > Schema > OBJECT > Tables > PERSOANA. The 'PERSOANA' table is highlighted. On the right, the 'Object Tools' pane shows the 'General' tab for the 'PERSOANA' table. The table is located in the 'OBJECT' schema and 'USERS' tablespace. The 'Table' type is 'Standard', and 'Use Abstract Datatype' is checked. The 'Columns' section shows the following structure:

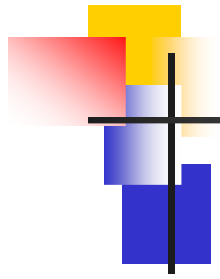
Name	Schema	Datatype	Size	Scale	Ref	Nulls?	Default Value	Scope Schema	Scope Tab
ID	<None>	NUMBER	4	0		✓			
NUME	OBJECT	NUME_OBJ				✓			
ADR_REF	OBJECT	ADR_OBJ			✓	✓		OBJECT	ADR_TAB



## Dangling REF

---

- Este posibil ca un obiect identificat prin REF să devină indisponibil ( dacă este șters sau dacă i se modifică privilegiile)
  - o astfel de referință se numește *dangling*.
  - dialectul Oracle SQL conține un operator (IS DANGLING) care face testarea REF-urilor relativ la această condiție.



# Dereferențierea referinței

- Accesarea unui obiect referit printr-un REF se numește dereferențierea respectivului REF.
  - În Oracle există operatorul Deref care realizează acest lucru.
- Dereferențierea unui *dangling REF* întoarce un obiect null
- În Oracle există și posibilitatea *dereferențierii implicite* a REF-urilor.

# Exemplu

- Dacă X este un obiect de tip PERSON, atunci expresia SQL:  
x.manager.name;
- urmărește pointerul de la  
persoana X către o alta persoana  
–managerul lui X, și regăsește  
numele managerului.

The screenshot displays the Oracle Enterprise Manager Console interface. At the top, a window titled 'SQL Scratchpad - obiect@ORACLE09' shows the following SQL code:

```
CREATE TYPE person AS OBJECT (  
  name VARCHAR2(30),  
  manager REF person );
```

The main console window, titled 'Oracle Enterprise Manager Console, Standalone', shows a tree view on the left with the following structure:

- Sequences
- Clusters
- Source Types
- User Types
  - Array Types
  - Object Types
    - ADR\_OBJ
    - ANG\_T
    - NUME\_OBJ
    - PERSON**
  - Table Types
  - XML Schema
- ODM
- ODM\_MTR

The right pane shows the 'General' tab for the 'PERSON' object type. The 'Name' field is set to 'PERSON' and the 'Schema' field is set to 'OBJECT'. Below the 'Attributes' section, there is a table with the following data:

Name	Datatype
NAME	VARCHAR2(30)
MANAGER	REF OBJECT.PERSON

The 'Methods' section is currently empty.

# Obținerea REF-urilor

Se poate obține un REF către un obiect rând prin selectarea acestuia din obiectul tabel care îl conține și aplicarea operatorului REF

