

CURS 3

SQL II



Funcțiile predefinite: COUNT, SUM, AVG, MAX, MIN

- Formatul general al unei fraze SELECT ce conține funcții predefinite:

SELECT *fcț_pred1, fcț_pred2,..fcț_predn*

FROM listă de tabele

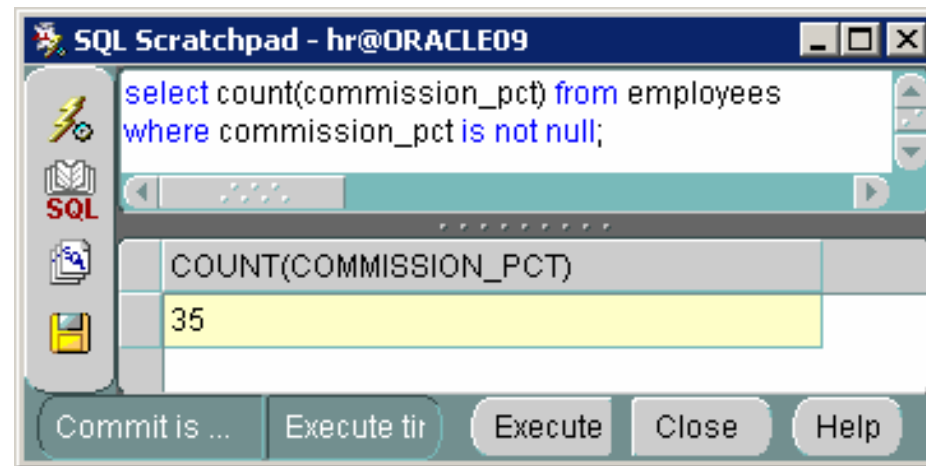
WHERE condiții

- În lipsa opțiunii GROUP BY, dacă în clauza SELECT este prezentă o funcție predefinită, tabelul rezultat va conține o singură linie.

Funcția COUNT

Contorizează valorile unei coloane - numără, într-o relație câte valori diferite de null are coloana specificată.

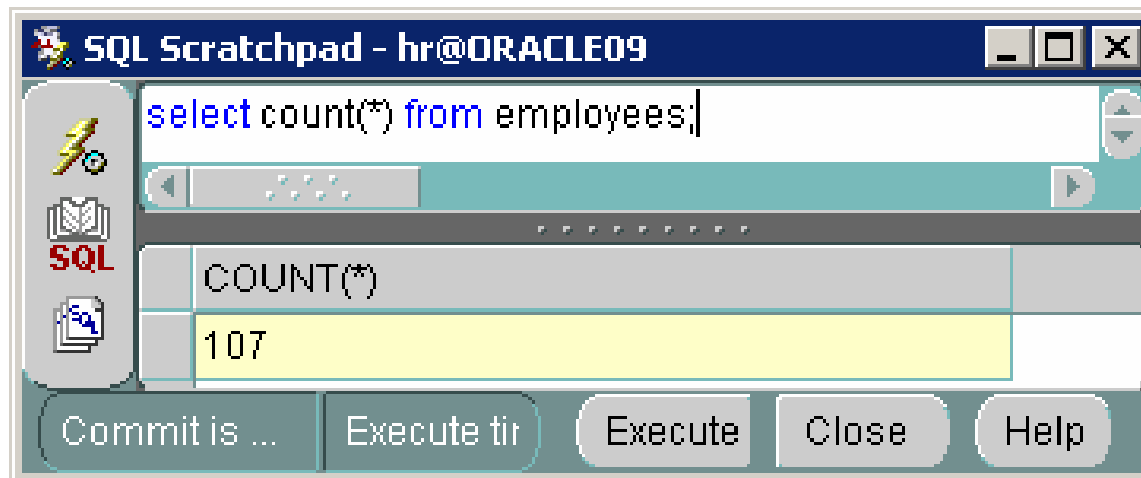
Câte comisioane se platesc?



Funcția COUNT

Câți angajati are firma?

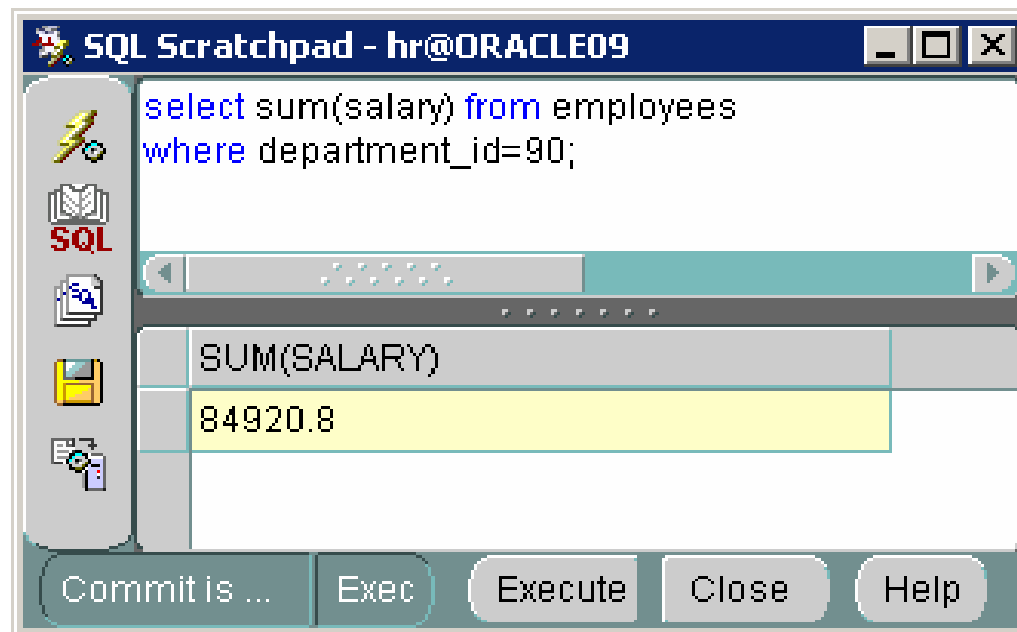
- se poate utiliza ca argument în locul numelui unei coloane semnul *;
 - se va determina câte linii are tabelul la care se aplică funcția respectivă



Funcția SUM

Calculează suma valorilor unei coloane.

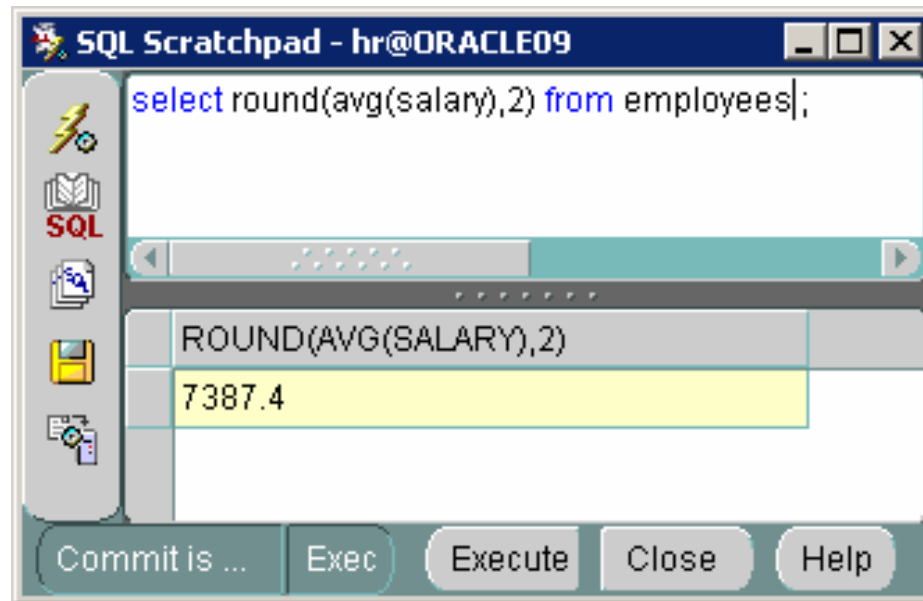
Care este valoarea totală a salariilor platite in departamentul 90?



Funcția AVG

Calculează media aritmetică a unei coloane într-un tabel oarecare

- *Care este valoarea medie a salariilor platite in firma?*



Gruparea tuplurilor.

Clauzele **GROUP BY** și **HAVING**

GROUP BY permite formarea grupurilor de tupluri într-o relație pe baza valorilor comune ale unei coloane.

- asocierea unei clauze **HAVING** la o clauză **GROUP BY** face posibilă selectarea anumitor grupe de tupluri care îndeplinesc un criteriu.
- Formatul general al clauzei **GROUP BY** este:

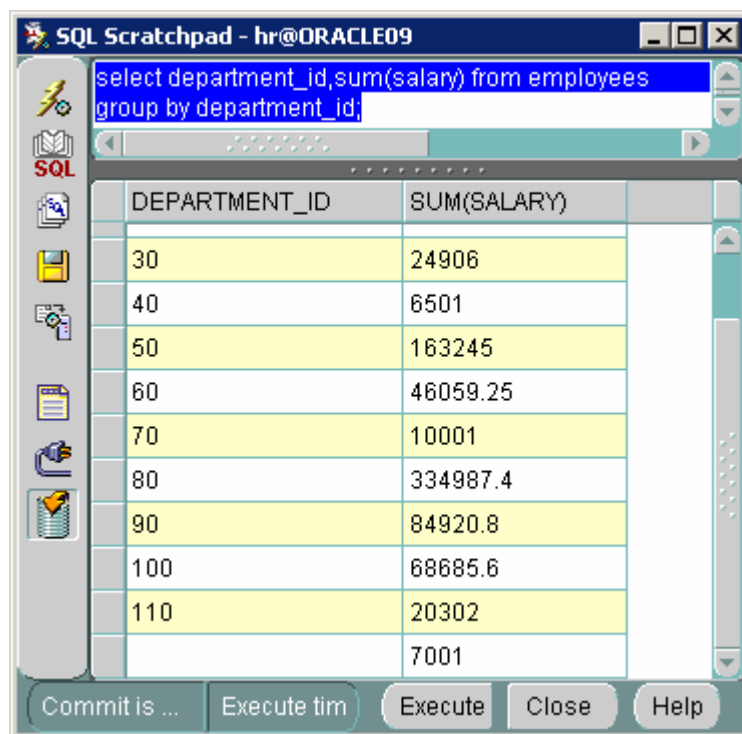
`SELECT col1, col2,...coln`

`FROM tabel`

`GROUP BY coloană de grupare`

Gruparea tuplurilor. Clauzele GROUP BY și HAVING

Care este totalul salariilor platite in fiecare departament?



The screenshot shows a window titled "SQL Scratchpad - hr@ORACLE09". The query entered is "select department_id,sum(salary) from employees group by department_id;". The results are displayed in a table with two columns: DEPARTMENT_ID and SUM(SALARY). The table contains 10 rows of data, with the last row showing a total sum of 7001.

DEPARTMENT_ID	SUM(SALARY)
30	24906
40	6501
50	163245
60	46059.25
70	10001
80	334987.4
90	84920.8
100	68685.6
110	20302
	7001

tabelul rezultat va aveaun număr de linii egal cu numărul departamentelor.



Clauza HAVING

Permite introducerea unor restricții care sunt aplicate grupurilor de tupluri, deci nu tuplurilor individuale, așa cum acționează clauza WHERE. Din tabelul rezultat sunt eliminate toate tuplurile care nu satisfac condiția dată.

- Clauza HAVING lucrează împreună cu o clauză GROUP BY, fiind practic o clauză WHERE aplicată acesteia. Formatul general este:

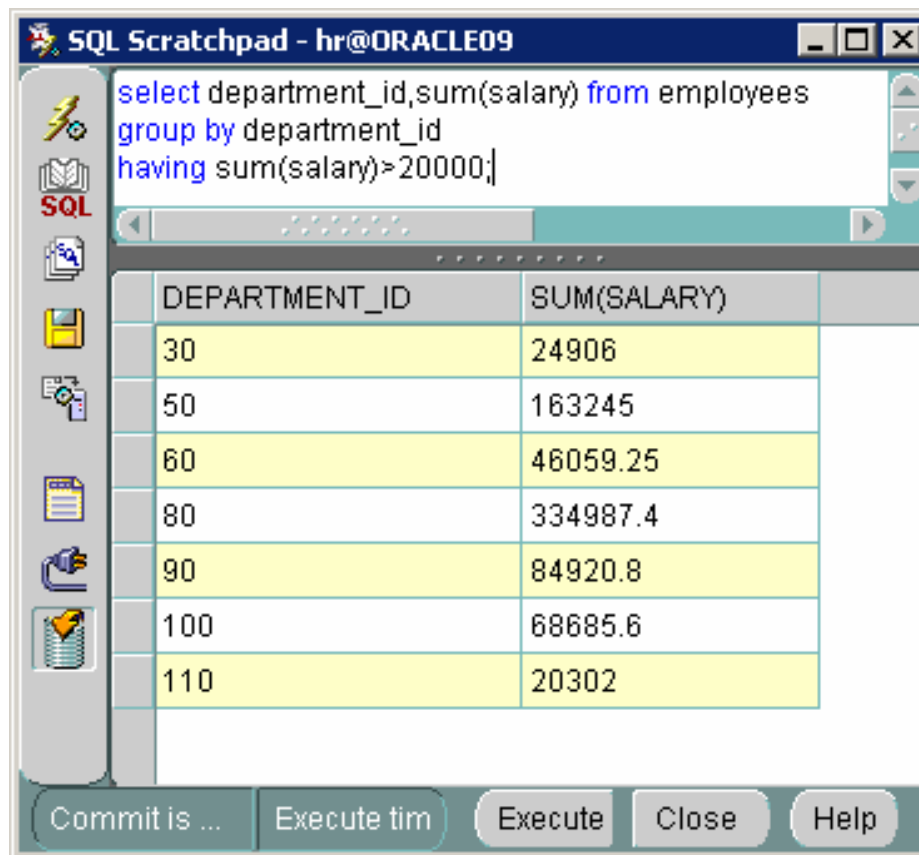
SELECT col1, col2,..coln

FROM tabel

GROUP BY coloană de regrupare

HAVING caracteristică de grup

Pentru salariile platite, interesează valoarea pe fiecare departament, numai dacă această valoare este mai mare de 20000.



The screenshot shows a window titled "SQL Scratchpad - hr@ORACLE09". The query entered is: `select department_id,sum(salary) from employees group by department_id having sum(salary)>20000;`. The results are displayed in a table with two columns: DEPARTMENT_ID and SUM(SALARY). The table contains seven rows of data, with alternating yellow and white background colors for each row.

DEPARTMENT_ID	SUM(SALARY)
30	24906
50	163245
60	46059.25
80	334987.4
90	84920.8
100	68685.6
110	20302

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



Actualizarea datelor în SQL

- implică trei acțiuni posibile :
 - adăugarea de noi linii la cele existente într-un tabel
 - ștergerea unor linii din tabel
 - modificarea valorii unor attribute.



Adăugarea datelor în SQL - INSERT

**INSERT INTO *NumeTab* [(*NumeCâmp1* [, *NumeCâmp2*, ...])]
VALUES (*eExpr1* [, *eExpr2*, ...])**

**INSERT INTO *NumeTab* FROM ARRAY *Nume* | FROM
MEMVAR**

- Comanda INSERT poate fi asociată cu o subinterogare, care să furnizeze valorile care trebuie adăugate prin copiere dintr-un alt tabel

**INSERT INTO *NumeTab1* [(*NumeCâmp1* [, *NumeCâmp2*, ...])]
SELECT [(*NumeCâmp1* [, *NumeCâmp2*, ...])]
FROM *NumeTab2*
WHERE *conditii***

- Subinterogarea se poate utiliza în locul unui nume de tabel în clauza INTO a comenzii INSERT:

**INSERT INTO
(SELECT [(*NumeCâmp1* [, *NumeCâmp2*, ...])]
FROM *NumeTab2*)
VALUES (*eExpr1* [, *eExpr2*, ...])**

Adăugarea datelor în SQL – Exemplul 1

SQL Scratchpad - hr@ORACLE09

```
insert into jobs(job_id, job_title, min_salary, max_salary)
values ('IT_CONS', 'IT Consultant', 5000, 10000);

select * from jobs where job_id like '%IT%';
```

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
IT_PROG	Programmer	4000	10000
IT_CONS	IT Consultant	5000	10000

Commit is ... Execute time (s): 0.0 Rows returned: 2 Execute Close Help

SQL Scratchpad - hr@ORACLE09

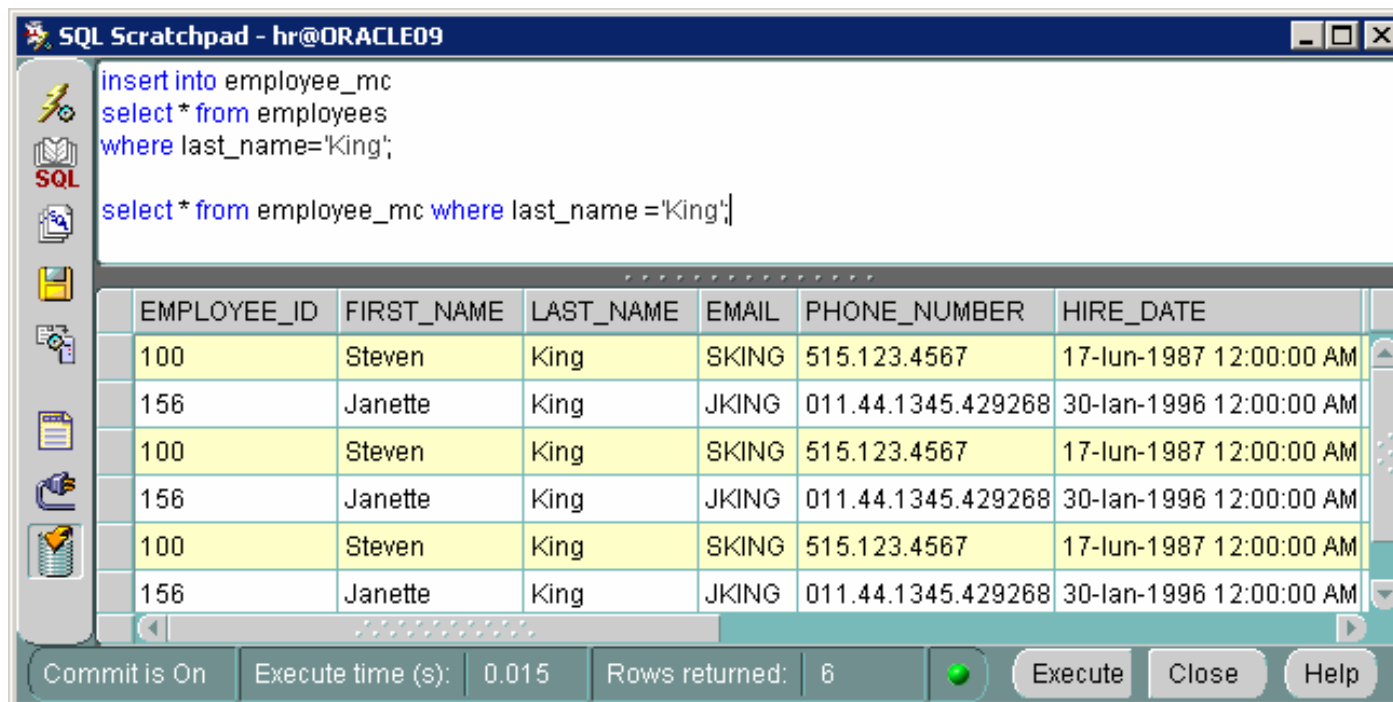
```
insert into jobs
values ('IT_DEV', 'IT Developer', 5000, 10000);

select * from jobs where job_id like '%IT%';
```

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
IT_PROG	Programmer	4000	10000
IT_CONS	IT Consultant	5000	10000
IT_DEV	IT Developer	5000	10000

Commit is ... Execute time (s): 0.0 Rows returned: 3 Execute Close Help

Adăugarea datelor în SQL – Exemplul 2



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

```
insert into employee_mc  
select * from employees  
where last_name='King';  
  
select * from employee_mc where last_name = 'King';
```

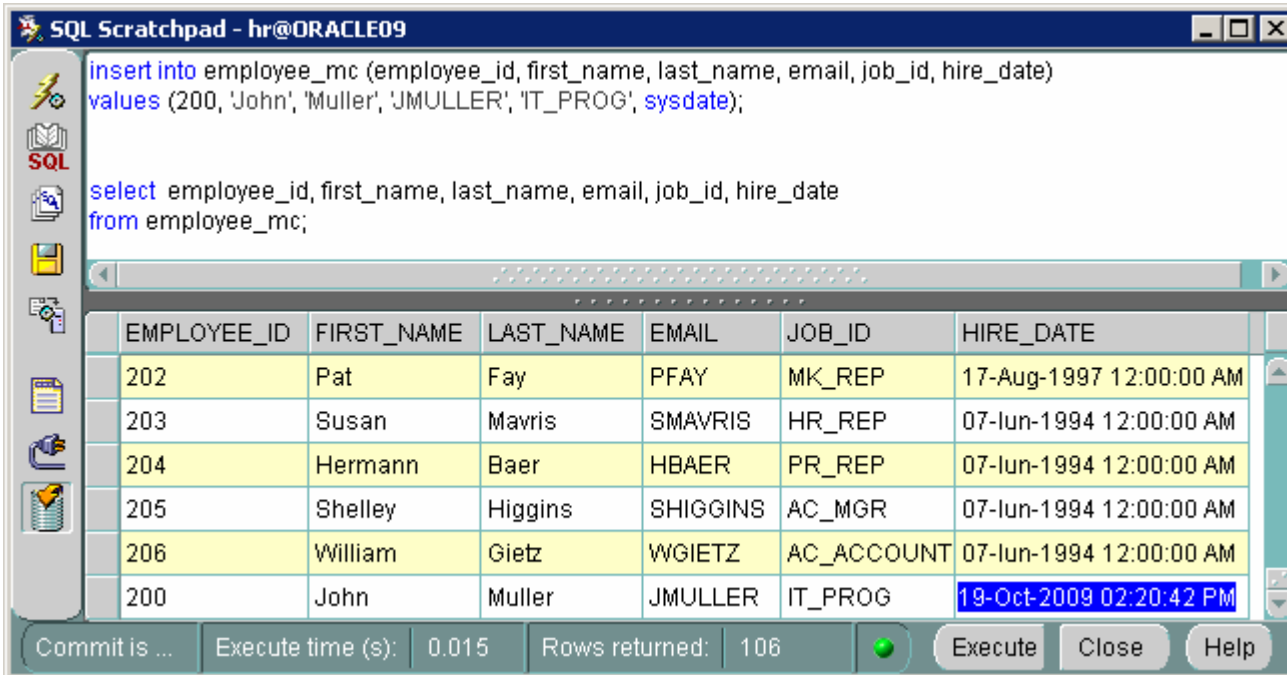
Below the editor, a table displays the results of the second query. The table has 7 columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, and HIRE_DATE. It contains 6 rows of data, which are duplicates of the data from the 'employees' table where last_name is 'King'.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
100	Steven	King	SKING	515.123.4567	17-lun-1987 12:00:00 AM
156	Janette	King	JKING	011.44.1345.429268	30-lan-1996 12:00:00 AM
100	Steven	King	SKING	515.123.4567	17-lun-1987 12:00:00 AM
156	Janette	King	JKING	011.44.1345.429268	30-lan-1996 12:00:00 AM
100	Steven	King	SKING	515.123.4567	17-lun-1987 12:00:00 AM
156	Janette	King	JKING	011.44.1345.429268	30-lan-1996 12:00:00 AM

At the bottom of the window, the status bar shows: "Commit is On", "Execute time (s): 0.015", "Rows returned: 6", and buttons for "Execute", "Close", and "Help".

Adăugarea datelor în SQL – Exemplul 3

introducerea valorilor speciale



The image shows a screenshot of a software application titled "SQL Scratchpad - hr@ORACLE09". The application has a text area at the top containing two SQL statements: an insert statement and a select statement. Below the text area is a table displaying the results of the select statement. The table has six columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, JOB_ID, and HIRE_DATE. The table contains six rows of data. The last row, with EMPLOYEE_ID 200, has a HIRE_DATE of 19-Oct-2009 02:20:42 PM, which is highlighted in blue. At the bottom of the window, there is a status bar with fields for "Commit is ...", "Execute time (s): 0.015", "Rows returned: 106", and buttons for "Execute", "Close", and "Help".

```
insert into employee_mc (employee_id, first_name, last_name, email, job_id, hire_date)
values (200, 'John', 'Muller', 'JMULLER', 'IT_PROG', sysdate);

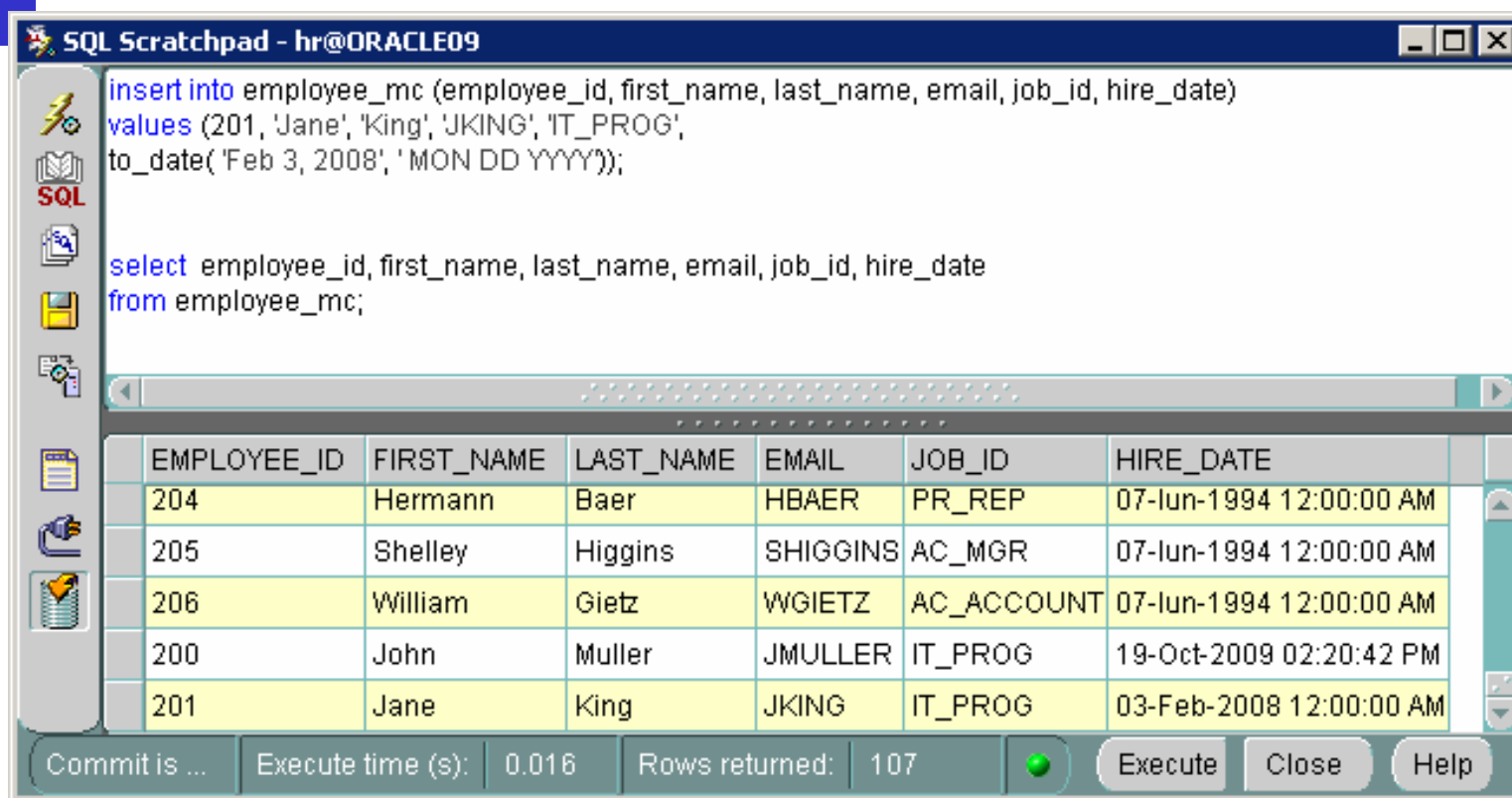
select employee_id, first_name, last_name, email, job_id, hire_date
from employee_mc;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	JOB_ID	HIRE_DATE
202	Pat	Fay	PFAY	MK_REP	17-Aug-1997 12:00:00 AM
203	Susan	Mavris	SMAVRIS	HR_REP	07-lun-1994 12:00:00 AM
204	Hermann	Baer	HBAER	PR_REP	07-lun-1994 12:00:00 AM
205	Shelley	Higgins	SHIGGINS	AC_MGR	07-lun-1994 12:00:00 AM
206	William	Gietz	WGIETZ	AC_ACCOUNT	07-lun-1994 12:00:00 AM
200	John	Muller	JMULLER	IT_PROG	19-Oct-2009 02:20:42 PM

Commit is ... Execute time (s): 0.015 Rows returned: 106 Execute Close Help

Adăugarea datelor în SQL – Exemplul 4

introducerea unei date calendaristice



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

```
insert into employee_mc (employee_id, first_name, last_name, email, job_id, hire_date)
values (201, 'Jane', 'King', 'JKING', 'IT_PROG',
to_date('Feb 3, 2008', 'MON DD YYYY'));

select employee_id, first_name, last_name, email, job_id, hire_date
from employee_mc;
```

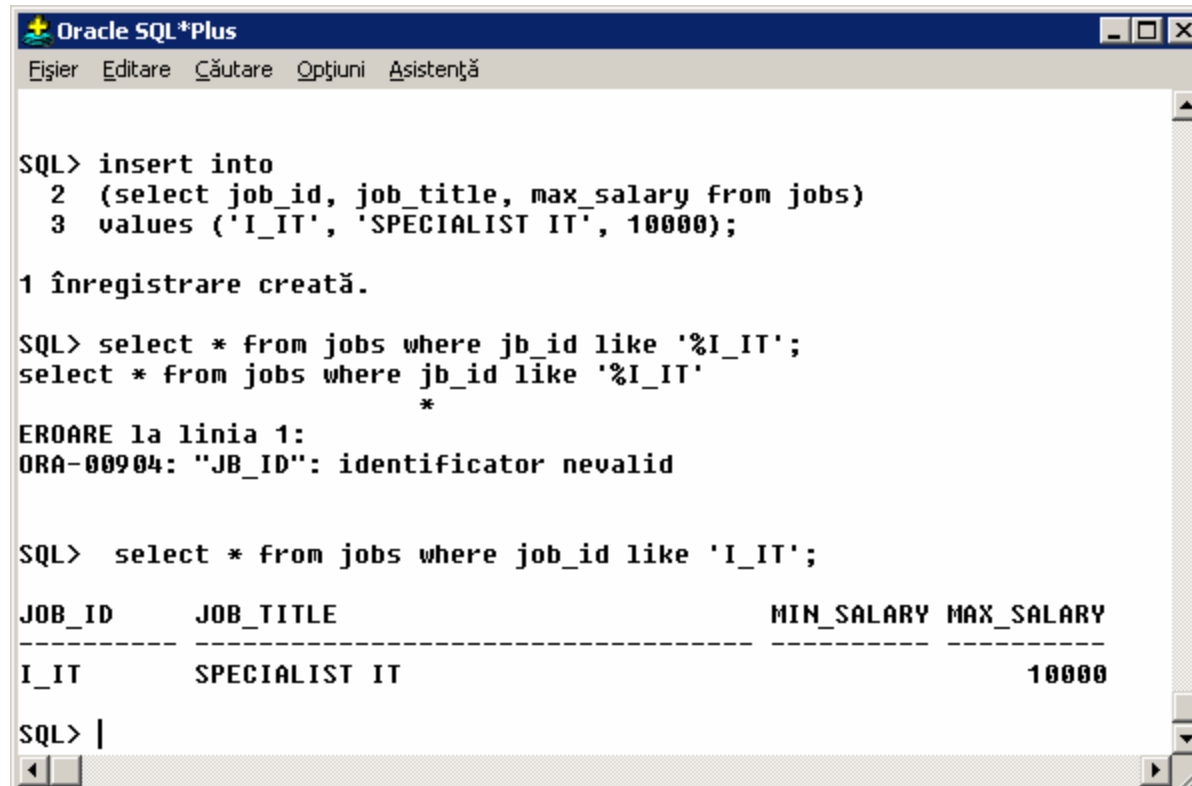
Below the editor, a table of results is displayed:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	JOB_ID	HIRE_DATE
204	Hermann	Baer	HBAER	PR_REP	07-lun-1994 12:00:00 AM
205	Shelley	Higgins	SHIGGINS	AC_MGR	07-lun-1994 12:00:00 AM
206	William	Gietz	WGIETZ	AC_ACCOUNT	07-lun-1994 12:00:00 AM
200	John	Muller	JMULLER	IT_PROG	19-Oct-2009 02:20:42 PM
201	Jane	King	JKING	IT_PROG	03-Feb-2008 12:00:00 AM

At the bottom of the window, a status bar shows: "Commit is ...", "Execute time (s): 0.016", "Rows returned: 107", and buttons for "Execute", "Close", and "Help".

Adăugarea datelor în SQL – Exemplul 5

Utilizarea subconsultarii pentru specificarea tabelului unde se face inserarea



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

SQL> insert into
  2 (select job_id, job_title, max_salary from jobs)
  3 values ('I_IT', 'SPECIALIST IT', 10000);

1 înregistrare creată.

SQL> select * from jobs where jb_id like '%I_IT';
select * from jobs where jb_id like '%I_IT'
                                *
EROARE la linia 1:
ORA-00904: "JB_ID": identificator nevalid

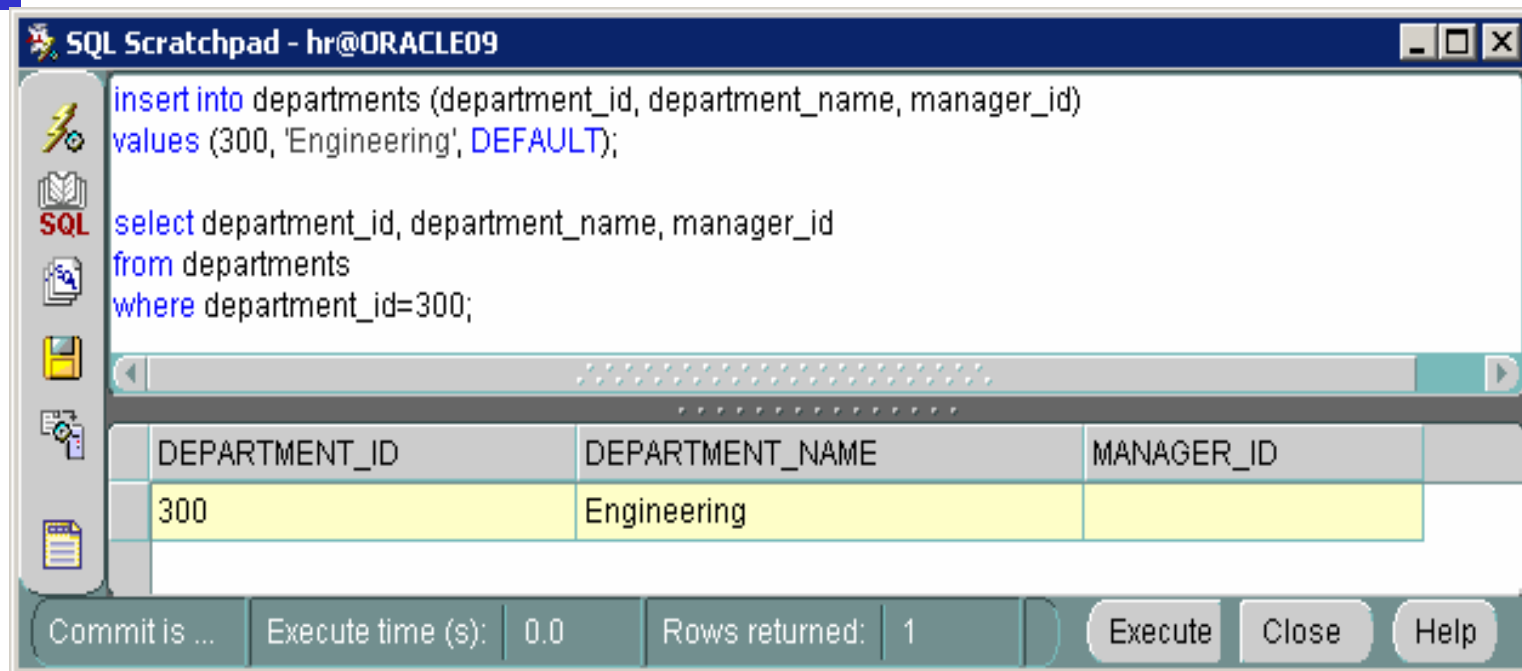
SQL> select * from jobs where job_id like 'I_IT';

JOB_ID      JOB_TITLE      MIN_SALARY MAX_SALARY
-----
I_IT        SPECIALIST IT      10000

SQL> |
```

Adăugarea datelor în SQL – Exemplul 6

Utilizarea valorilor implicite



The screenshot shows a window titled "SQL Scratchpad - hr@ORACLE09". The window contains two SQL queries and a table of results.

```
insert into departments (department_id, department_name, manager_id)
values (300, 'Engineering', DEFAULT);

select department_id, department_name, manager_id
from departments
where department_id=300;
```

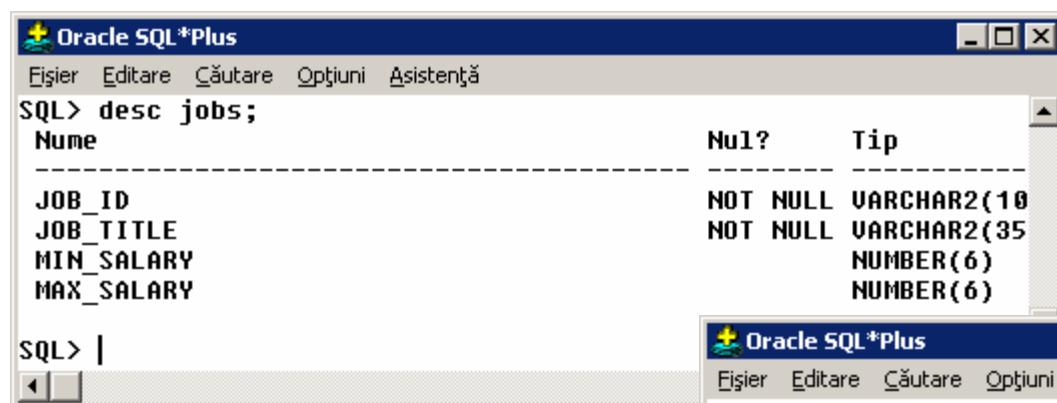
DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID
300	Engineering	

At the bottom of the window, there is a status bar with the following information:

- Commit is ...
- Execute time (s): 0.0
- Rows returned: 1
- Buttons: Execute, Close, Help

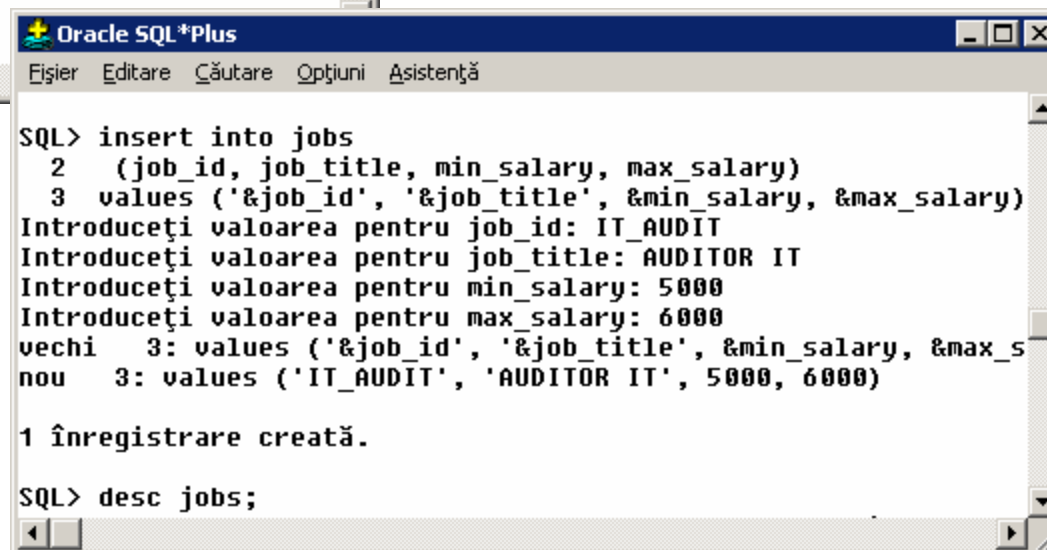
Crearea unui script -1

Presupune posibilitatea introducerii de la tastatura a valorilor ce se doresc inserate



```
Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență
SQL> desc jobs;
Nume                                Nu1?    Tip
-----
JOB_ID                             NOT NULL VARCHAR2(10
JOB_TITLE                           NOT NULL VARCHAR2(35
MIN_SALARY                          NUMBER(6)
MAX_SALARY                          NUMBER(6)

SQL> |
```



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

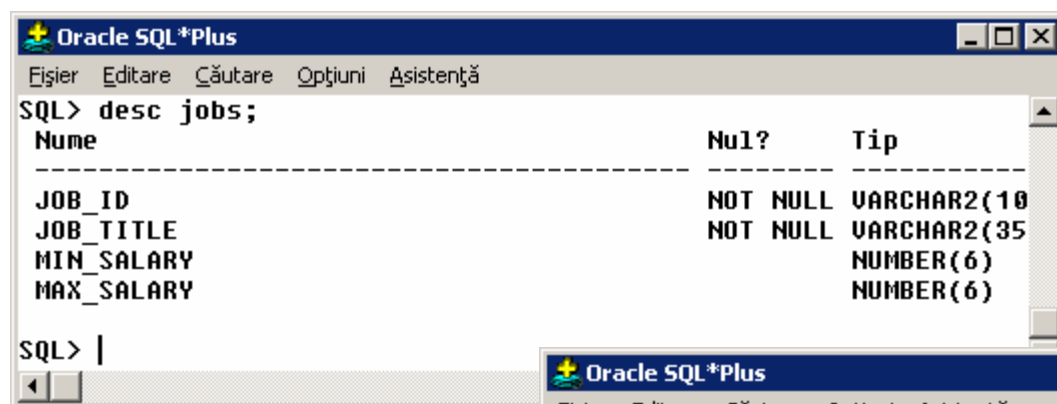
SQL> insert into jobs
  2  (job_id, job_title, min_salary, max_salary)
  3  values ('&job_id', '&job_title', &min_salary, &max_salary)
Introduceți valoarea pentru job_id: IT_AUDIT
Introduceți valoarea pentru job_title: AUDITOR IT
Introduceți valoarea pentru min_salary: 5000
Introduceți valoarea pentru max_salary: 6000
vechi  3: values ('&job_id', '&job_title', &min_salary, &max_s
nou    3: values ('IT_AUDIT', 'AUDITOR IT', 5000, 6000)

1 înregistrare creată.

SQL> desc jobs;
```

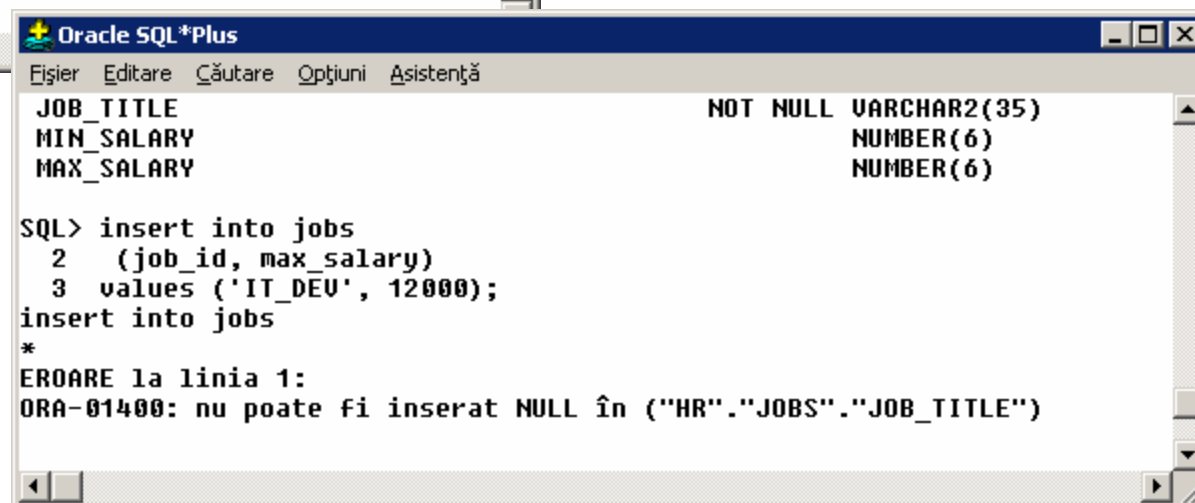
Crearea unui script -2

Ce se intampla daca se omite introducerea unei valori in campul job_title?



```
Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență
SQL> desc jobs;
Nume                                Nul?    Tip
-----
JOB_ID                             NOT NULL VARCHAR2(10)
JOB_TITLE                           NOT NULL VARCHAR2(35)
MIN_SALARY                           NUMBER(6)
MAX_SALARY                           NUMBER(6)

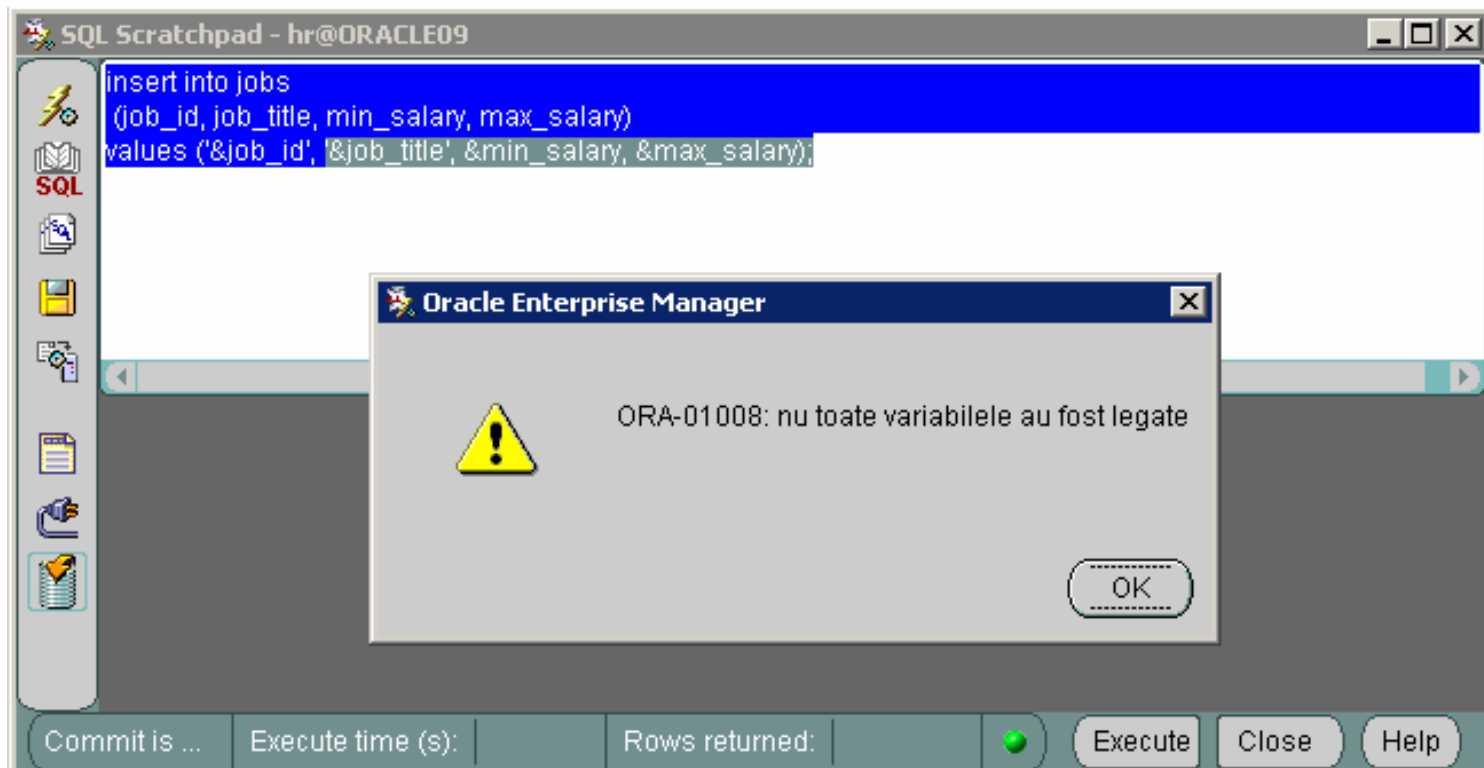
SQL> |
```



```
Oracle SQL*Plus
Fișier Editare Căutare Opțiuni Asistență
JOB_TITLE                           NOT NULL VARCHAR2(35)
MIN_SALARY                           NUMBER(6)
MAX_SALARY                           NUMBER(6)

SQL> insert into jobs
      2  (job_id, max_salary)
      3  values ('IT_DEU', 12000);
insert into jobs
*
EROARE la linia 1:
ORA-01400: nu poate fi inserat NULL în ("HR"."JOBS"."JOB_TITLE")
```

Crearea unui script -3 utilizarea SQL Scratchpad





Stergerea de linii din tabel - DELETE

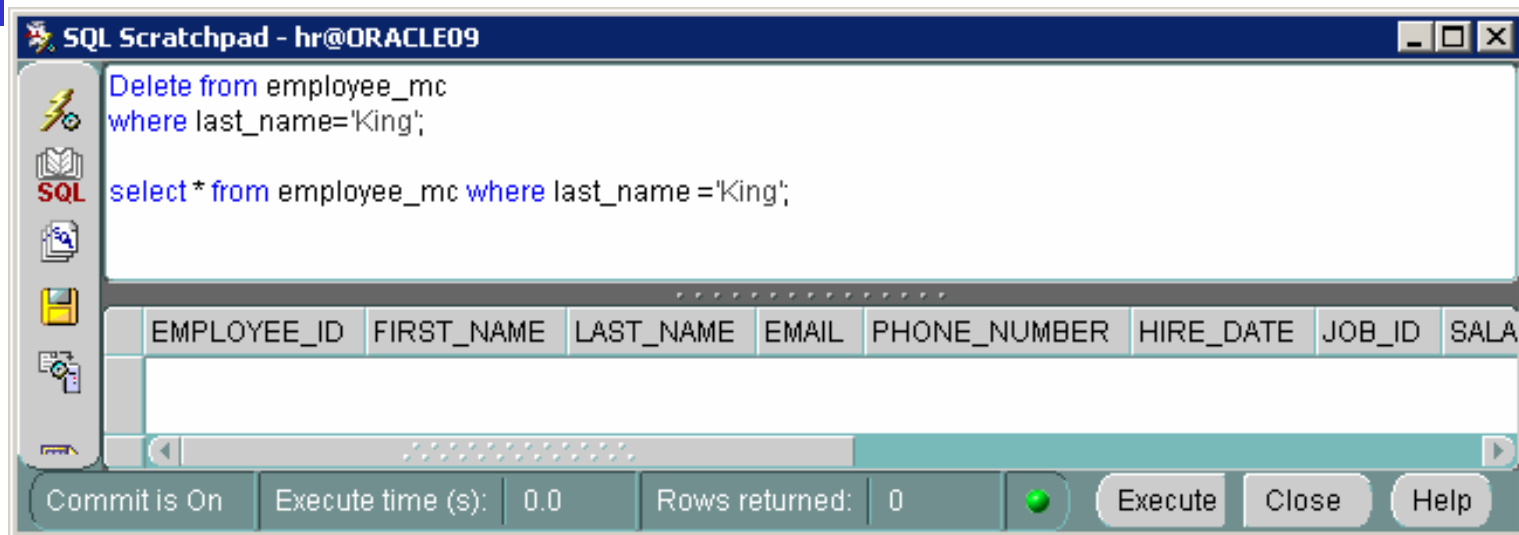
```
DELETE FROM [NumeBD!]NumeTab  
    [WHERE CondFiltru1 [AND | OR CondFiltru2 ...]]
```

Sau în combinație cu subinterogari

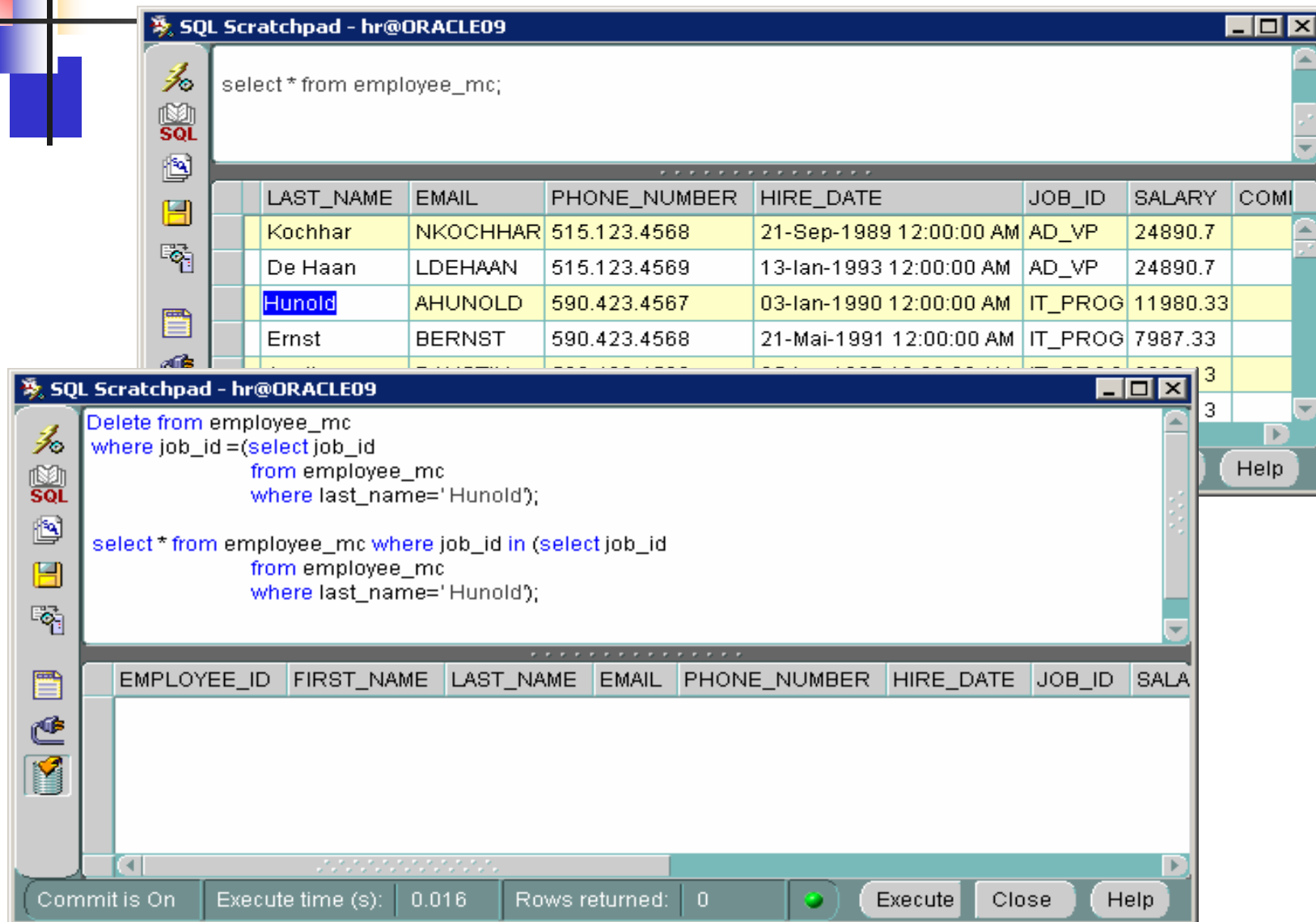
```
DELETE FROM [NumeBD!]NumeTab  
    [WHERE câmp operator  
        (SELECT câmp  
        FROM NumeTabel  
        WHERE condiții)]
```

```
DELETE FROM facturi  
WHERE cod =  
    (SELECT cod  
    FROM agenti  
    WHERE nume = "ELCO S.A.");
```

Stergerea de linii din tabel - DELETE



Stergerea de linii din tabel - DELETE



The image shows two windows of the SQL Scratchpad application. The top window displays the result of a query: `select * from employee_mc;`. The bottom window displays the SQL code for deleting a record from the `employee_mc` table based on the `last_name` 'Hunold'.

Top Window: SQL Scratchpad - hr@ORACLE09

```
select * from employee_mc;
```

LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMI
Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989 12:00:00 AM	AD_VP	24890.7	
De Haan	LDEHAAN	515.123.4569	13-Jan-1993 12:00:00 AM	AD_VP	24890.7	
Hunold	AHUNOLD	590.423.4567	03-Jan-1990 12:00:00 AM	IT_PROG	11980.33	
Ernst	BERNST	590.423.4568	21-Mai-1991 12:00:00 AM	IT_PROG	7987.33	

Bottom Window: SQL Scratchpad - hr@ORACLE09

```
Delete from employee_mc
where job_id=(select job_id
              from employee_mc
              where last_name='Hunold');

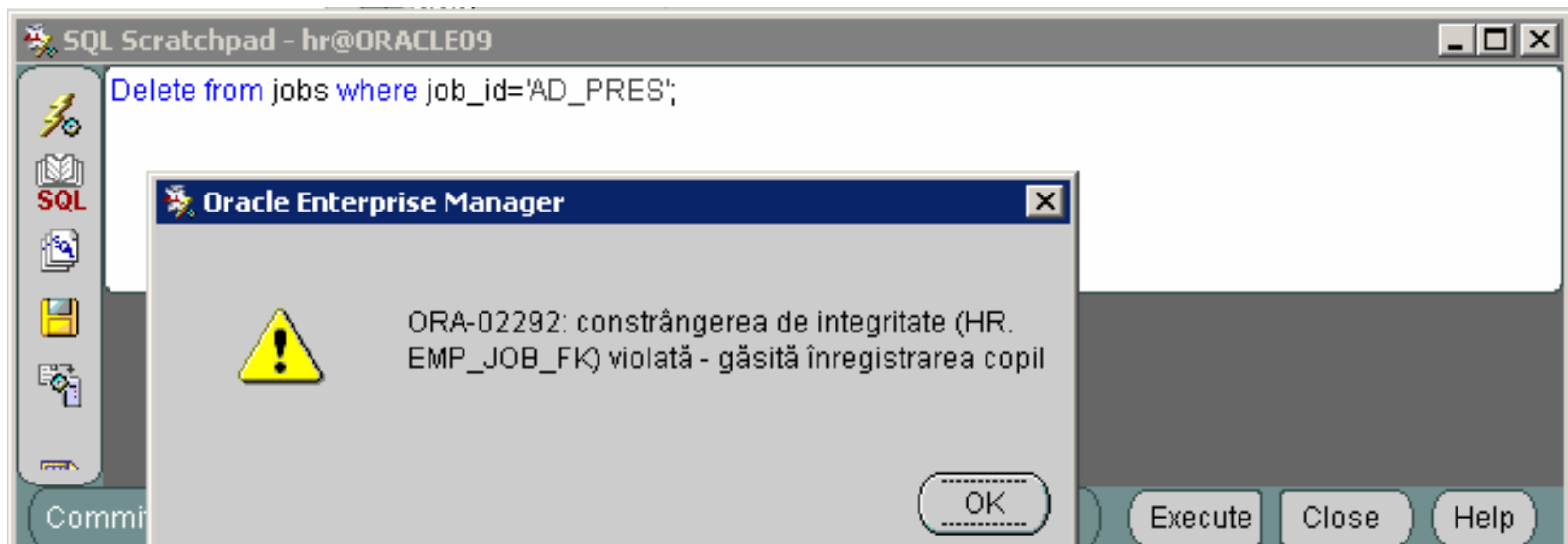
select * from employee_mc where job_id in (select job_id
                                           from employee_mc
                                           where last_name='Hunold');
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALA
-------------	------------	-----------	-------	--------------	-----------	--------	------

Commit is On | Execute time (s): 0.016 | Rows returned: 0 | Execute | Close | Help

Stergerea de linii din tabel - DELETE

*Sunt variante SQL care, la crearea unui tabel, permit descrierea acțiunii care se va derula la ștergerea unei linii (restricționare sau stergere cascadată). **DECE???***





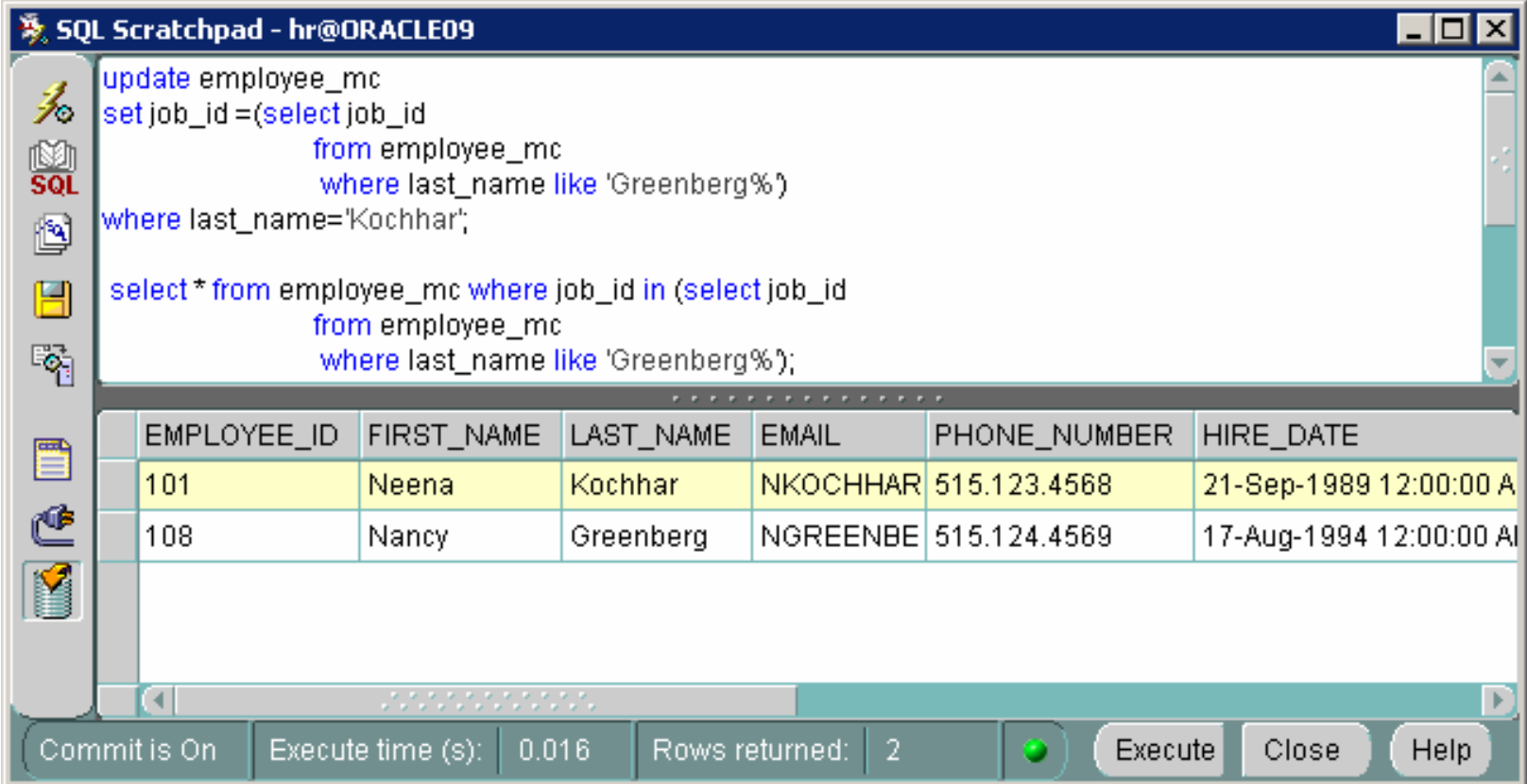
Actualizarea valorilor câmpurilor - UPDATE

```
UPDATE [NumeBD1!]NumeTabel  
    SET NumeCâmp1 = Expr1  
    [, NumeCâmp2 = Expr2 ...]  
    WHERE CondFiltru1 [AND | OR CondFiltru2 ...]]
```

Sau folosind subinterogările

```
UPDATE NumeTabel  
    SET NumeCâmp1 = (SELECT NumeCâmp1  
                      FROM NumeTabel  
                      WHERE condiții),  
    [NumeCâmp2 = (SELECT NumeCâmp2  
                  FROM NumeTabel  
                  WHERE condiții),]  
WHERE condiții
```

Actualizarea valorilor câmpurilor - UPDATE



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

```
update employee_mc
set job_id=(select job_id
            from employee_mc
            where last_name like 'Greenberg%')
where last_name='Kochhar';

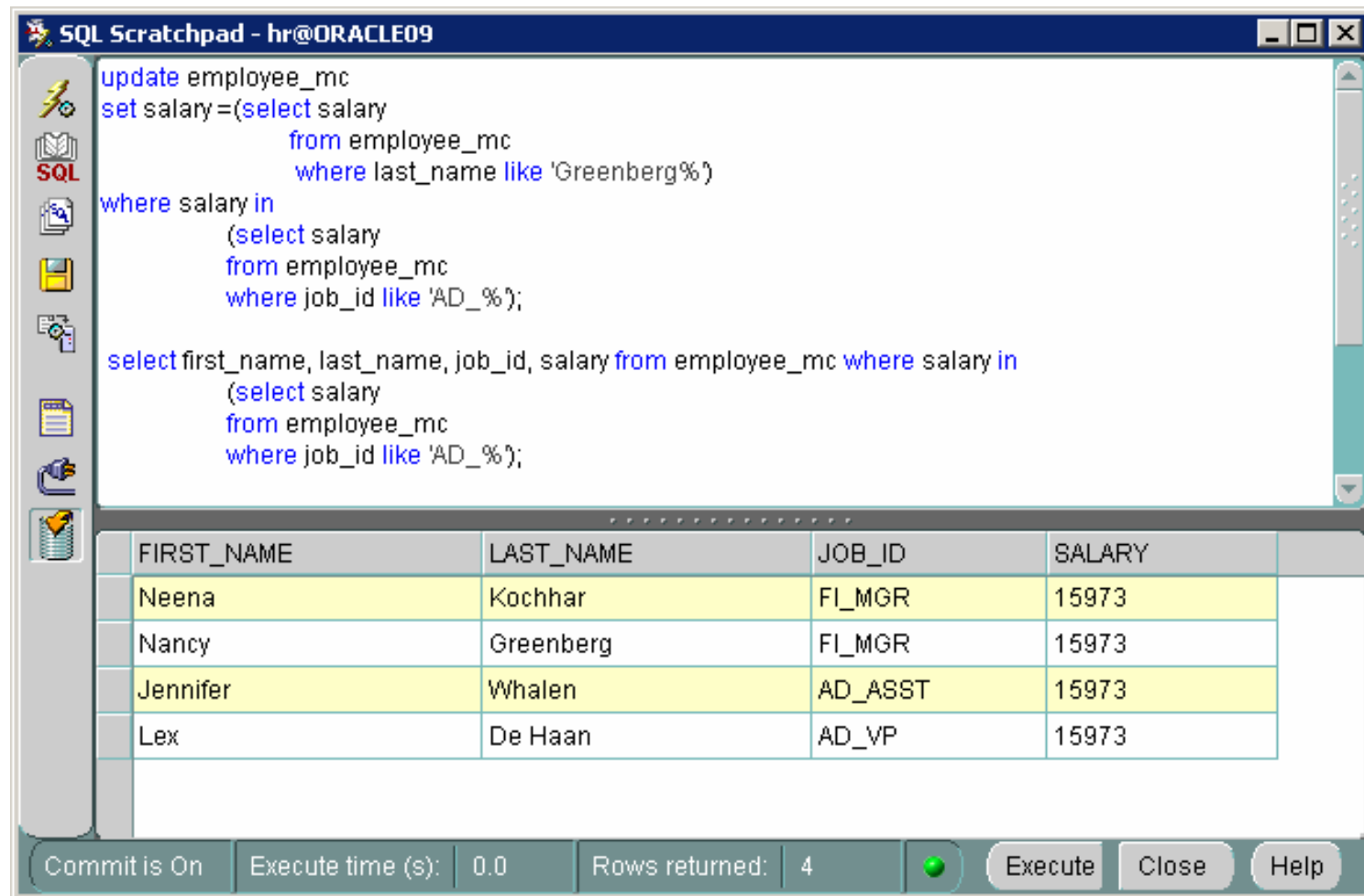
select * from employee_mc where job_id in (select job_id
                                           from employee_mc
                                           where last_name like 'Greenberg%');
```

Below the editor, a table displays the results of the second query:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-1989 12:00:00 A
108	Nancy	Greenberg	NGREENBE	515.124.4569	17-Aug-1994 12:00:00 A

At the bottom, a status bar shows "Commit is On", "Execute time (s): 0.016", "Rows returned: 2", and buttons for "Execute", "Close", and "Help".

Actualizarea valorilor câmpurilor - UPDATE



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

```
update employee_mc
set salary=(select salary
            from employee_mc
            where last_name like 'Greenberg%')
where salary in
(select salary
 from employee_mc
 where job_id like 'AD_%');

select first_name, last_name, job_id, salary from employee_mc where salary in
(select salary
 from employee_mc
 where job_id like 'AD_%');
```

Below the code, a table displays the results of the query. The table has four columns: FIRST_NAME, LAST_NAME, JOB_ID, and SALARY. It contains four rows of data:

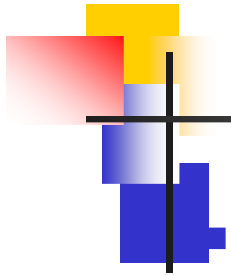
FIRST_NAME	LAST_NAME	JOB_ID	SALARY
Neena	Kochhar	FI_MGR	15973
Nancy	Greenberg	FI_MGR	15973
Jennifer	Whalen	AD_ASST	15973
Lex	De Haan	AD_VP	15973

At the bottom of the window, a status bar shows "Commit is On", "Execute time (s): 0.0", "Rows returned: 4", and buttons for "Execute", "Close", and "Help".



Vederi

- *vedere* este o relație virtuală- o relație care nu este de fapt de sine stătătoare, ci este derivată, în mod dinamic din una sau mai multe relații de bază
- în realitate, nu există în baza de date - este produsă la un moment dat la cererea unui anumit utilizator
- *relația de bază* este o relație cu o anumită denumire, corespunzătoare unei entități din schema conceptuală, ale cărei tupluri sunt stocate fizic în baza de date.
- vederea este rezultatul dinamic al uneia sau mai multor operații relaționale, care acționează asupra relațiilor de bază pentru a realiza o altă relație



Vederile

- sunt dinamice
- furnizează un mecanism de securitate puternic și flexibil
 - ascunderea unor părți ale bazei de date față de anumiți utilizatori,
- permit utilizatorilor accesarea datelor într-un mod personalizat, conform cerințelor lor și
- pot simplifica operațiunile complexe asupra relațiilor de bază.

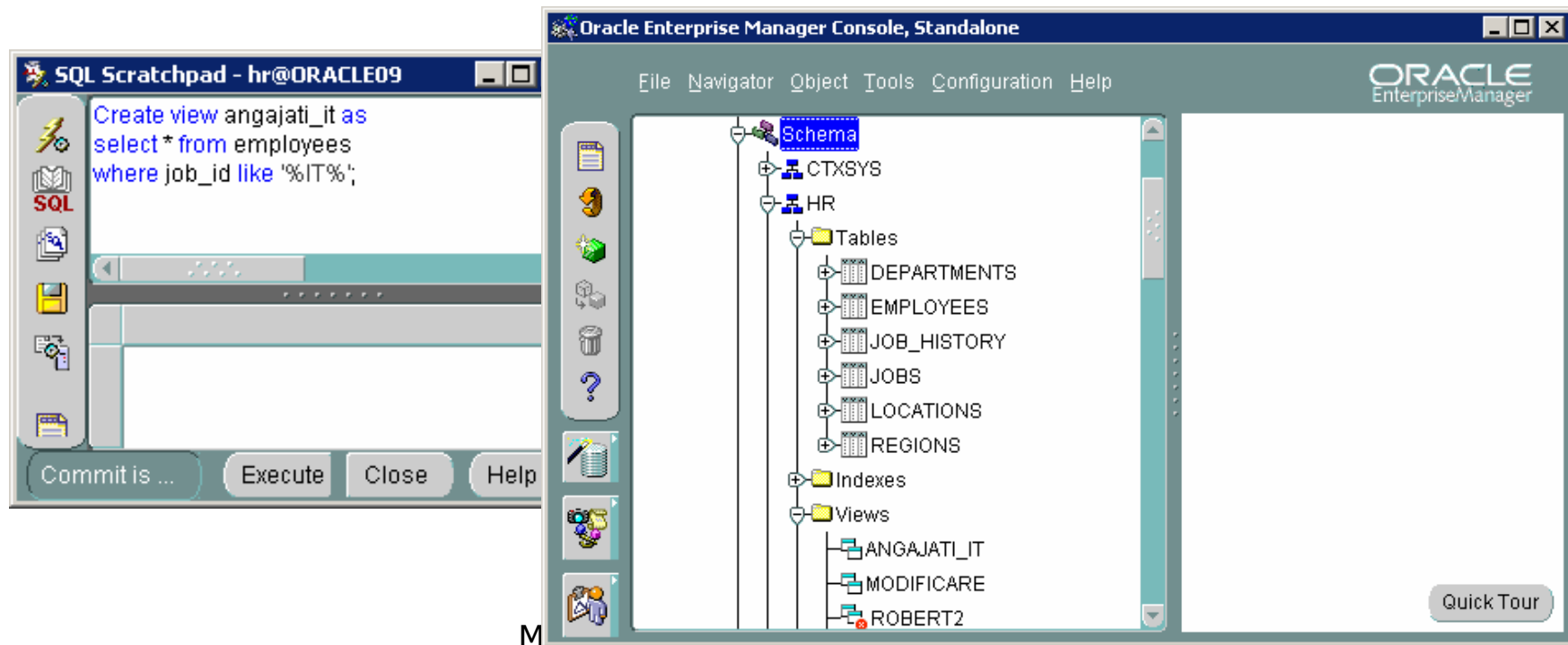


Reactualizarea vederilor...

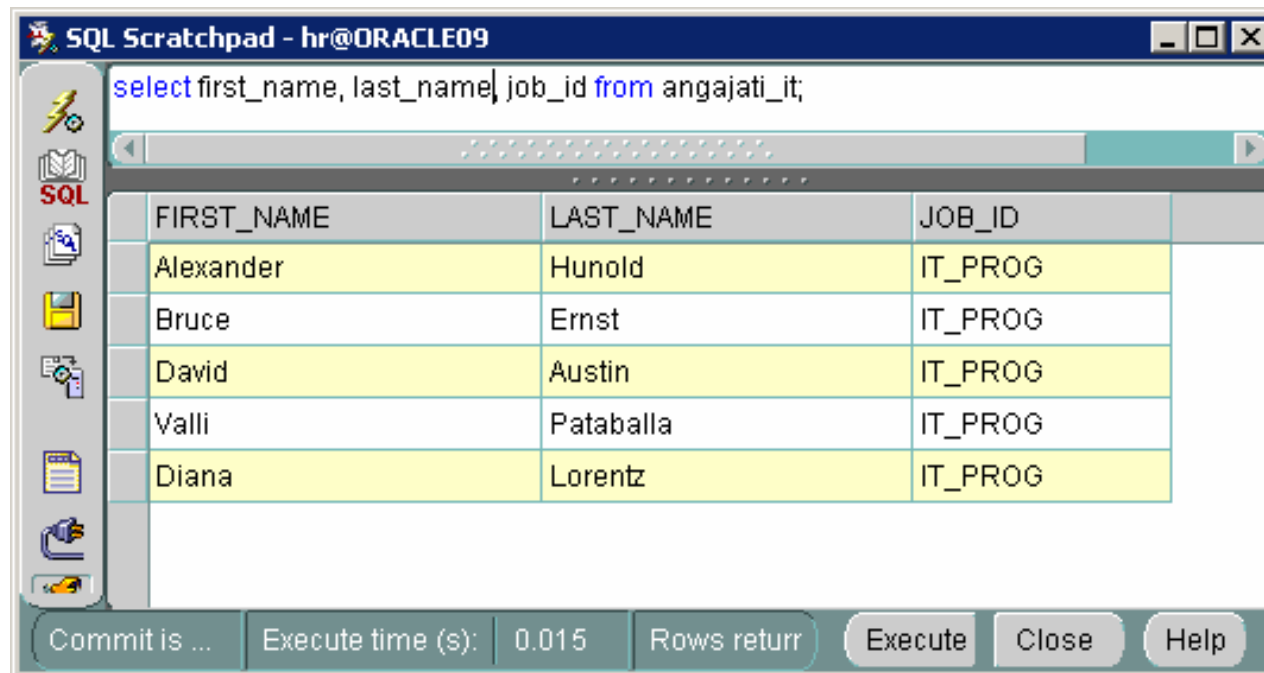
- ...cu reflectarea corespunzătoare a acestei reactualizări în relațiile de bază:
 - este permisă prin intermediul unei vederi definite prin utilizarea unei interogări simple, care implică o singură relație de bază și conține fie cheia primară fie cheia candidat a acesteia;
 - nu este permisă prin vederi care implică relații de bază multiple;
 - nu este permisă prin vederi care implică operații de acumulare sau de grupare.

Crearea vederilor

```
CREATE SQL VIEW [NumeVedere] [REMOTE]  
[CONNECTION NumeConexiune] [SHARE]  
| CONNECTION NumeSursaDate  
[AS SQL SELECT Statement]
```



Consultarea unei vederi



The screenshot shows a window titled "SQL Scratchpad - hr@ORACLE09". The query entered is `select first_name, last_name, job_id from angajati_it;`. The results are displayed in a table with three columns: FIRST_NAME, LAST_NAME, and JOB_ID. The table contains five rows of data. The status bar at the bottom shows "Commit is ...", "Execute time (s): 0.015", "Rows return", and buttons for "Execute", "Close", and "Help".

FIRST_NAME	LAST_NAME	JOB_ID
Alexander	Hunold	IT_PROG
Bruce	Ernst	IT_PROG
David	Austin	IT_PROG
Valli	Pataballa	IT_PROG
Diana	Lorentz	IT_PROG

Eliminarea unei vederi

DROP VIEW *NumeVedere* [RESTRICT|CASCADE]

