

# Baze de date

## Limbajul SQL

**Teams: FI-AIA-2-Baze de date-2022/2023**



THE **INFORMATION** COMPANY

# *Curs 6*

## *Limbajul SQL*

# *Limbajul SQL*

## **6. Subinterogări (Subqueries)**

### **6.1. SINGLE ROW SUBQUERIES**

### **6.2. MULTIPLE ROW SUBQUERIES**

## 6. SUBQUERIES (Subinterogari)

În **SQL**, subinterogările ne permit să aflăm o informație care ne este necesară pentru a obține informația pe care o vrem.

- O **subinterogare (subquery)** este o instrucțiune **SELECT** care este inclusă în clauza unei alte instrucțiuni **SELECT**.

## 6. SUBQUERIES (Subinterogari)

- Subinterogarea poate fi plasata în una din următoarele clauze:
  - **WHERE**
  - **HAVING**
  - **FROM**
- **Subinterogarea** se execută prima dată, iar rezultatul este folosit pentru obținerea rezultatului de către interogarea principală (**outer query**).

## 6. SUBQUERIES (Subinterogari)

Sintaxa generală:

```
SELECT select_list  
FROM table  
WHERE expression operator  
                (SELECT select_list  
                FROM table);
```

# 6. SUBQUERIES (Subinterogari)

## Reguli de folosire a subinterogarilor

- O **subinterogare** se pune între paranteze rotunde
- O **subinterogare** este plasată în partea dreaptă a unei condiții de comparare
- Interogarea exterioară și **subinterogarea**-ul pot prelua date din tabele diferite

## 6. SUBQUERIES (Subinterogari)

- Într-o instrucțiune **SELECT** se poate folosi o singură clauză **ORDER BY** și, dacă se folosește, trebuie să fie ultima clauza a interogării principale.
- Un subquery nu poate avea propria clauză **ORDER BY**.
- Singura limită a numărului de interogări este dimensiunea buffer-ului folosit de interogare.
- Dacă subinterogarea returnează **null** sau nu returnează nici o linie, atunci interogarea exterioară nu va returna nimic.



## 6. SUBQUERIES (Subinterogari)

Sunt două tipuri de subinterogări(subqueries):

- 1) **single-row subqueries** – care folosesc operatorii single-row: **>, =, >=, <, <=** și dau ca rezultat o **singură linie**.
- 2) **multiple-row subqueries** – care folosesc operatorii multiple-row: **IN, ANY, ALL** și dau ca rezultat **mai multe linii**.

# *Limbajul SQL*

## **6. SUBQUERIES (Subinterogări)**

### **6.1. SINGLE ROW SUBQUERIES**

### **6.2. MULTIPLE ROW SUBQUERIES**

## 6.1. SINGLE ROW SUBQUERIES

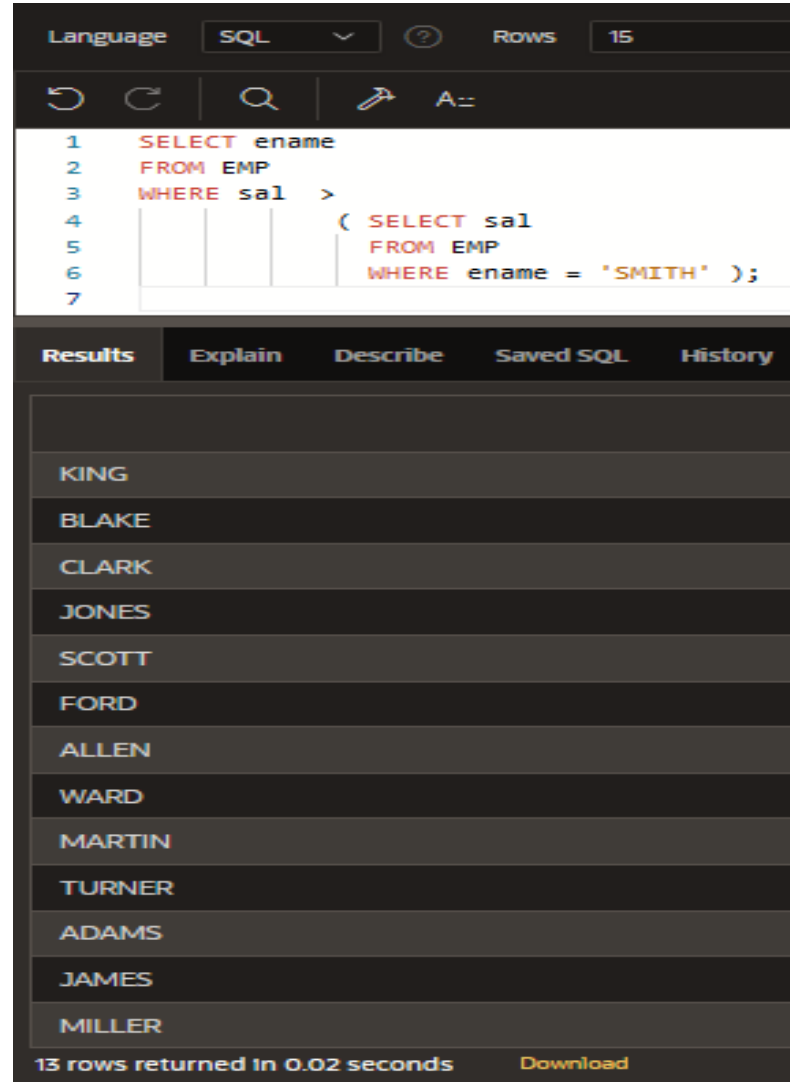
### Single row-subquery

```
SELECT ename  
FROM EMP  
WHERE sal >  
      ( SELECT sal  
        FROM EMP  
        WHERE ename = 'SMITH' );
```

Aflati numele angajatilor  
care au salariul mai mare  
decat angajatul care se  
numeste SMITH.

# 6.1. SINGLE ROW SUBQUERIES

## Single row-subquery



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', a search icon, and 'Rows' set to '15'. Below the toolbar, the SQL query is displayed in a text editor with line numbers 1 through 7. The query is a single row subquery that selects the names of employees whose salary is greater than that of the employee named 'SMITH'. The results are shown in a table below the query, with columns for the employee names. The table contains 13 rows of data. At the bottom of the results table, it says '13 rows returned in 0.02 seconds' and there is a 'Download' button.

```
1 SELECT ename
2 FROM EMP
3 WHERE sal >
4     ( SELECT sal
5       FROM EMP
6       WHERE ename = 'SMITH' );
7
```

KING
BLAKE
CLARK
JONES
SCOTT
FORD
ALLEN
WARD
MARTIN
TURNER
ADAMS
JAMES
MILLER

13 rows returned in 0.02 seconds [Download](#)

## 6.1. SINGLE ROW SUBQUERIES

### Subcereri din mai multe tabele

Subcererile (subinterogările) nu sunt limitate la o singură interogare (cerere).

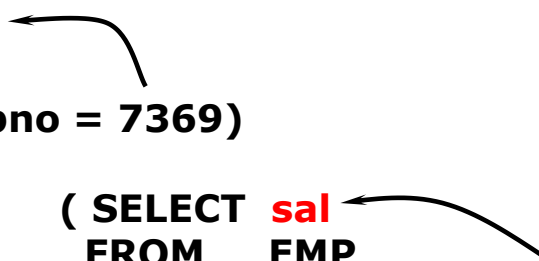
Așa cum se poate observa în exemplul următor, pot fi mai mult de o singură interogare.

De asemenea se pot face interogări din tabele diferite.

Exemplul urmator afiseaza angajatii a caror functie este acelasi cu cel al angajatului cu numarul 7369 si a caror salariu este mai mare decat cel al angajatului 7875.

## Executarea unei subinterogari single-row

```
SELECT ename, job
FROM EMP
WHERE job =
  (SELECT job
   FROM EMP
   WHERE empno = 7369)
AND sal >
  ( SELECT sal
    FROM EMP
    WHERE empno = 7876 );
```



ENAME	JOB
-----	-----
MILLER	CLERK

## 6.1. SINGLE ROW SUBQUERIES

- Exemplul este format din 3 blocuri de cereri:
  - o cerere exterioara
  - doua cereri interne
- Blocurile de cereri interne sunt primele executate, producand rezultatele cererii: FUNCTIONAR (CLERK), respectiv 1300.
- Blocul exterior de cereri este apoi procesat si foloseste valorile returnate de catre cererile interne pentru a finaliza propriile conditii de cautare.
- Ambele cereri interne returneaza valori singulare (FUNCTIONAR si 1300), astfel ca aceasta instructiune SQL este denumita o subinterogare single-row.

# 6.1. SINGLE ROW SUBQUERIES

Language SQL ? Rows 15 ? Clear Command Find Tables

↶ ↷ 🔍 📌 A±

```
1  SELECT ename, job
2  FROM EMP
3  WHERE job =
4      (SELECT job
5       FROM EMP
6       WHERE empno = 7369)
7  AND sal >
8      ( SELECT sal
9       FROM EMP
10      WHERE empno = 7876 );
11
```

**Results** Explain Describe Saved SQL History

ENAME	JOB
MILLER	CLERK

1 rows returned in 0.01 seconds Download



## 6.1. SINGLE ROW SUBQUERIES

**SELECT** ename, job, sal, deptno

**FROM** EMP

**WHERE** job =

(**SELECT** job

**FROM** EMP

**WHERE** empno = 7934 )

**AND** deptno =

(**SELECT** deptno

**FROM** DEPT

**WHERE** dname = 'ACCOUNTING');

# 6.1. SINGLE ROW SUBQUERIES

```
1  SELECT ename, job, sal, deptno
2  FROM EMP
3  WHERE job =
4      (SELECT job
5       FROM EMP
6       WHERE empno = 7934 )
7  AND deptno =
8      (SELECT deptno
9       FROM dept
10      WHERE dname = 'ACCOUNTING');
11
```

Results Explain Describe Saved SQL History			
ENAME	JOB	SAL	DEPTNO
MILLER	CLERK	1300	10

## 6.1. SINGLE ROW SUBQUERIES

Se pot folosi funcțiile de grup în subinterogări.

O funcție de grup utilizată în subquery fără clauza **GROUP BY**, returnează o singură linie.

**SELECT** nume, prenume, salariu

**FROM** angajati

**WHERE** salariu <

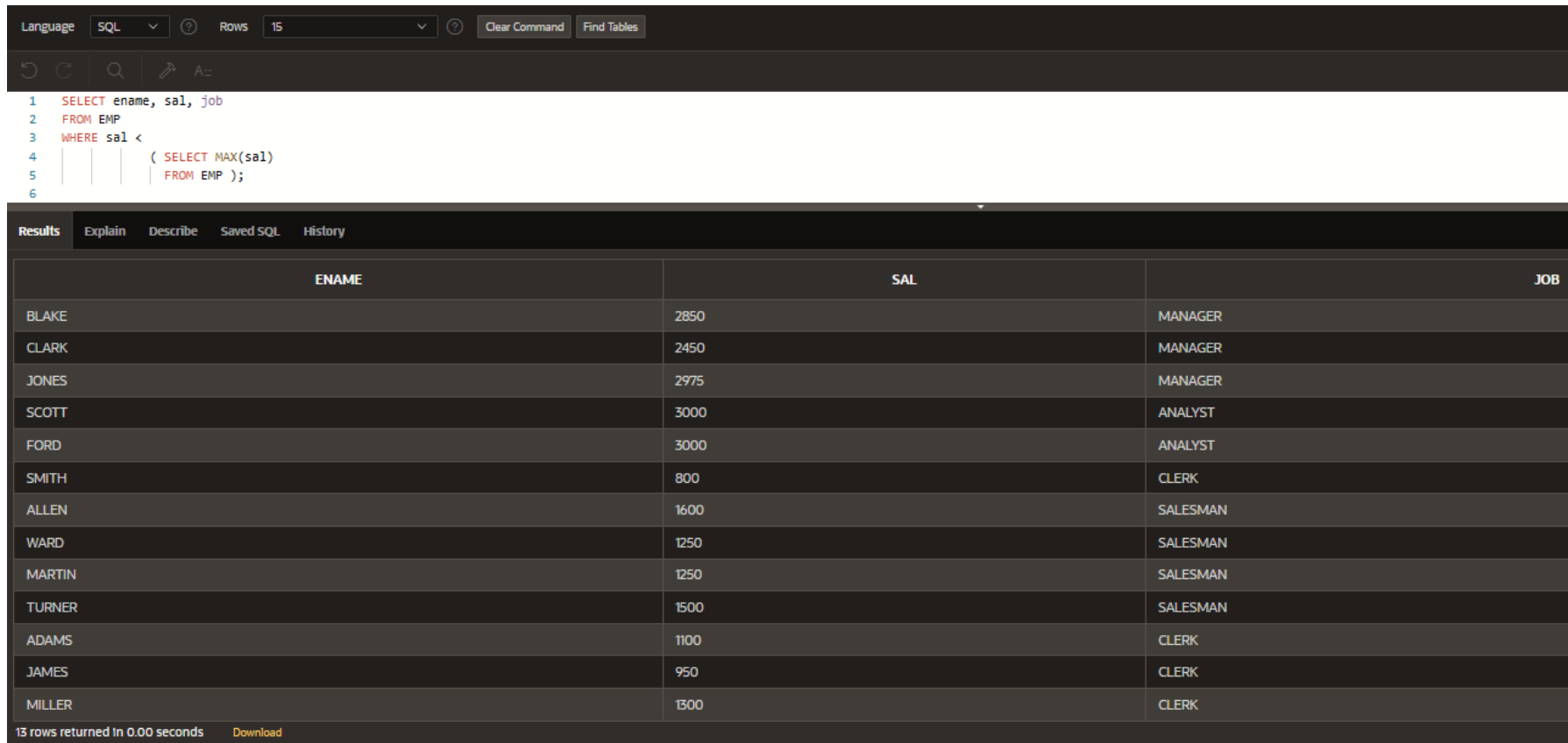
(**SELECT** **MAX**(salariu)

**FROM** angajati);

# 6.1. SINGLE ROW SUBQUERIES

Se pot folosi funcțiile de grup în subinterogări.

O funcție de grup utilizată în subquery fără clauza **GROUP BY**, returnează o singură linie.



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '15', and buttons for 'Clear Command' and 'Find Tables'. Below the toolbar is a search bar. The main area displays a SQL query:

```
1 SELECT ename, sal, job
2 FROM EMP
3 WHERE sal <
4     ( SELECT MAX(sal)
5       FROM EMP );
6
```

Below the query, there's a 'Results' tab with sub-tabs for 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with 13 rows. The table has three columns: 'ENAME', 'SAL', and 'JOB'. The data is as follows:

ENAME	SAL	JOB
BLAKE	2850	MANAGER
CLARK	2450	MANAGER
JONES	2975	MANAGER
SCOTT	3000	ANALYST
FORD	3000	ANALYST
SMITH	800	CLERK
ALLEN	1600	SALESMAN
WARD	1250	SALESMAN
MARTIN	1250	SALESMAN
TURNER	1500	SALESMAN
ADAMS	1100	CLERK
JAMES	950	CLERK
MILLER	1300	CLERK

At the bottom of the results, it says '13 rows returned in 0.00 seconds' and there is a 'Download' button.

## 6.1. SINGLE ROW SUBQUERIES

- Subinterogările pot fi plasate și în clauza **HAVING**.
- Deoarece clauza **HAVING** are întotdeauna o condiție de grup, și subinterogarea va avea aproape întotdeauna o condiție de grup.

```
SELECT deptno, MIN(sal)
FROM EMP
GROUP BY deptno
HAVING MIN(sal) >
    ( SELECT MIN(sal)
      FROM EMP
      WHERE deptno = 20 );
```

# 6.1. SINGLE ROW SUBQUERIES

- Subinterogările pot fi plasate și în clauza **HAVING**.
- Deoarece clauza **HAVING** are întotdeauna o condiție de grup, și subinterogarea va avea aproape întotdeauna o condiție de grup.

The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '15', and buttons for 'Clear Command' and 'Find Tables'. Below the toolbar is a command area with a search icon and a text input field. The SQL query is as follows:

```
1 SELECT deptno, MIN(sal)
2 FROM EMP
3 GROUP BY deptno
4 HAVING MIN(sal) >
5     (SELECT MIN(sal)
6      FROM EMP
7      WHERE deptno = 20);
8
```

Below the query, the 'Results' tab is active, showing a table with two columns: 'DEPTNO' and 'MIN(SAL)'. The table contains two rows of data:

DEPTNO	MIN(SAL)
30	950
10	1300

At the bottom of the results section, it says '2 rows returned in 0.01 seconds' and there is a 'Download' button.

# 6.1. SINGLE ROW SUBQUERIES

## Aplicatii rezolvate

- 1) Care este numele membrilor din personalul de la firma "COSTICA S.R.L.", al căror salariu este mai mare decât angajatul cu ID-ul 7698?
- 2) Care dintre angajatii **Oracle** au acelasi id al departamentului ca si cel corespunzator cu departamentul RESEARCH?

## 6.1. SINGLE ROW SUBQUERIES

- 1) Care este numele membrilor din personalul de la firma “COSTICA S.R.L.”, al căror salariu este mai mare decât angajatul cu ID-ul 7698?

```
SELECT ename  
FROM EMP  
WHERE sal >  
        ( SELECT sal  
          FROM EMP  
          WHERE empno = 7698 );
```



# 6.1. SINGLE ROW SUBQUERIES

The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '15', and buttons for 'Clear Command' and 'Find Tables'. Below the toolbar is a command area with a SQL query:

```
1 SELECT ename
2 FROM EMP
3 WHERE sal >
4     ( SELECT sal
5       FROM EMP
6       WHERE empno = 7698 );
7
```

Below the command area is a results panel with tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with one column, 'ENAME', and four rows of data:

ENAME
KING
JONES
SCOTT
FORD

At the bottom of the results panel, it says '4 rows returned in 0.01 seconds' and has a 'Download' button.

## 6.1. SINGLE ROW SUBQUERIES

2) Care dintre angajatii **Oracle** au acelasi id al departamentului ca si cel corespunzator cu departamentul RESEARCH?

```
SELECT ename, sal  
FROM EMP  
WHERE deptno =  
        ( SELECT deptno  
          FROM DEPT  
          WHERE dname = 'RESEARCH' );
```

# 6.1. SINGLE ROW SUBQUERIES

Language SQL ? Rows 15 ? Clear Command Find Tables

↶ ↷ 🔍 🔗 A..

```
2 FROM EMP
3 WHERE deptno =
4 ( SELECT deptno
5   FROM DEPT
6   WHERE dname = 'RESEARCH' );
7
```

**Results** Explain Describe Saved SQL History

ENAME	SAL
JONES	2975
SCOTT	3000
FORD	3000
SMITH	800
ADAMS	1100

5 rows returned in 0.00 seconds Download

# *Limbajul SQL*

## **6. SUBQUERIES (Subinterogări)**

### **6.1. SINGLE ROW SUBQUERIES**

### **6.2. MULTIPLE ROW SUBQUERIES**

## 6.2. MULTIPLE ROW SUBQUERIES

*Sunt acele subinterogări care dau ca rezultat mai multe valori.*

Folosesc operatorii **multiple row**:

1. **IN**
2. **ANY**
3. **ALL**

Operatorul **NOT** poate fi folosit în combinație cu oricare dintre aceștia.

## 6.2. MULTIPLE ROW SUBQUERIES

### 1. Operatorul IN

Operatorul **IN** este folosit dacă în interogarea exterioară clauza **WHERE** este folosită pentru a selecta acele valori care sunt egale cu una dintre valorile din lista returnată de subinterogare (**inner query**).

```
SELECT ename, sal, deptno  
FROM EMP  
WHERE sal IN  
    ( SELECT MIN(sal)  
      FROM EMP  
      GROUP BY deptno );
```

## 6.2. MULTIPLE ROW SUBQUERIES

- 1. Operatorul IN** - Operatorul **IN** este folosit dacă în interogarea exterioară clauza **WHERE** este folosită pentru a selecta acele valori care sunt egale cu una dintre valorile din lista returnată de subinterogare (**inner query**).

Language SQL ? Rows 15 ? Clear Command Find Tables

↶ ↷ 🔍 📌 A:

```
1 SELECT ename, sal, deptno
2 FROM EMP
3 WHERE sal IN
4   ( SELECT MIN(sal)
5     FROM EMP
6     GROUP BY deptno );
7
```

Results Explain Describe Saved SQL History

ENAME	SAL	DEPTNO
SMITH	800	20
JAMES	950	30
MILLER	1300	10

3 rows returned in 0.01 seconds Download

## 6.2. MULTIPLE ROW SUBQUERIES

### 2. Operatorul ANY

Acest operator este folosit atunci când dorim ca interogarea exterioară să selecteze valori egale, mai mici sau mai mari decât cel puțin o valoare dintre cele extrase de **subquery**.

**SELECT** ename, sal, hiredate

**FROM** EMP

**WHERE** hiredate < **ANY**

( **SELECT** hiredate

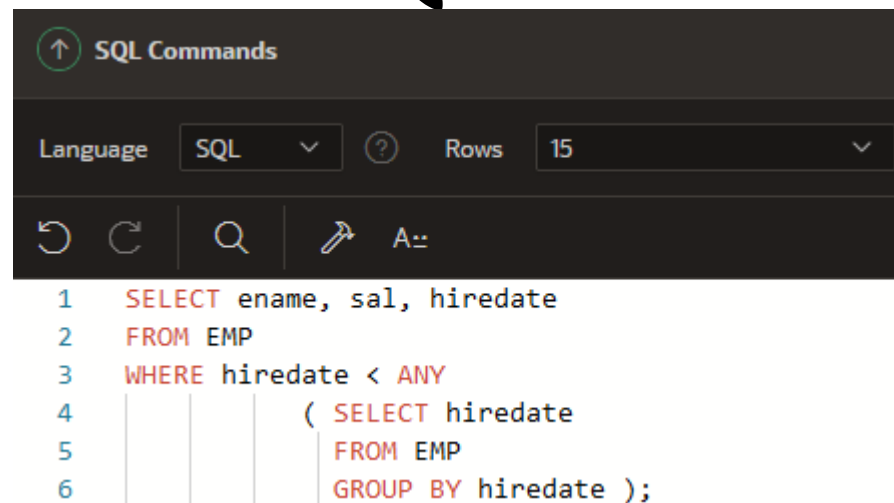
**FROM** EMP

**GROUP BY** hiredate );



## 6.2. MULTIPLE ROW SUBQUERIES

**2. Operatorul ANY** - Acest operator este folosit atunci când dorim ca interogarea exterioară să selecteze valori egale, mai mici sau mai mari decât cel puțin o valoare dintre cele extrase de **subquery**.



```

1  SELECT ename, sal, hiredate
2  FROM EMP
3  WHERE hiredate < ANY
4      ( SELECT hiredate
5        FROM EMP
6        GROUP BY hiredate );
  
```

Results			
<a href="#">Explain</a> <a href="#">Describe</a> <a href="#">Saved SQL</a> <a href="#">History</a>			
ENAME	SAL	HIREDATE	
SMITH	800	12/17/1980	
ALLEN	1600	02/20/1981	
WARD	1250	02/22/1981	
JONES	2975	04/02/1981	
BLAKE	2850	05/01/1981	
CLARK	2450	06/09/1981	
TURNER	1500	09/08/1981	
MARTIN	1250	09/28/1981	
KING	5000	11/17/1981	
FORD	3000	12/03/1981	
JAMES	950	12/03/1981	
MILLER	1300	01/23/1982	
SCOTT	3000	12/09/1982	

13 rows returned in 0.01 seconds [Download](#)

## 6.2. MULTIPLE ROW SUBQUERIES

### 3. Operatorul ALL

Acest operator este folosit atunci când dorim ca interogarea exterioară să selecteze valori egale, mai mici sau mai mari decât toate valorile extrase de subquery.

```
SELECT ename, sal, hiredate  
FROM EMP  
WHERE hiredate > ALL  
    ( SELECT hiredate  
      FROM EMP  
      GROUP BY hiredate );
```

## 6.2. MULTIPLE ROW SUBQUERIES

### VALORI NULL

Dacă una dintre valorile returnate de subinterogarea **multiple row** este **null**, dar celelalte valori nu sunt **null**, atunci:

- Dacă sunt folosiți operatorii **IN** sau **ANY**, interogarea exterioară va returna liniile care se potrivesc cu valorile **non-null**.
- Dacă este folosit operatorul **ALL**, interogarea exterioară nu va returna nimic.

## 6.2. MULTIPLE ROW SUBQUERIES

Clauzele **GROUP BY** și **HAVING**

- Pot fi folosite cu subinterogările de tip **multiple row**.

```
SELECT deptno, MIN(sal)
FROM EMP
GROUP BY deptno
HAVING MIN(sal) < ANY
      ( SELECT sal
        FROM EMP
        WHERE deptno IN (10,20) );
```

## 6.2. MULTIPLE ROW SUBQUERIES

### Clauzele **GROUP BY** și **HAVING**

- Pot fi folosite cu subinterogările de tip **multiple row**.

```
SQL Commands

Language SQL ? Rows 15 ?

1 SELECT deptno, MIN(sal)
2 FROM EMP
3 GROUP BY deptno
4 HAVING MIN(sal) < ANY
5 | | | | | | ( SELECT sal
6 | | | | | | FROM EMP
7 | | | | | | WHERE deptno IN (10,20) );
```

ResultsExplainDescribeSaved SQLHistory

DEPTNO	MIN(SAL)
30	950
10	1300
20	800

3 rows returned in 0.00 secondsDownload

## 6.2. MULTIPLE ROW SUBQUERIES

Clauzele **GROUP BY** si **HAVING**

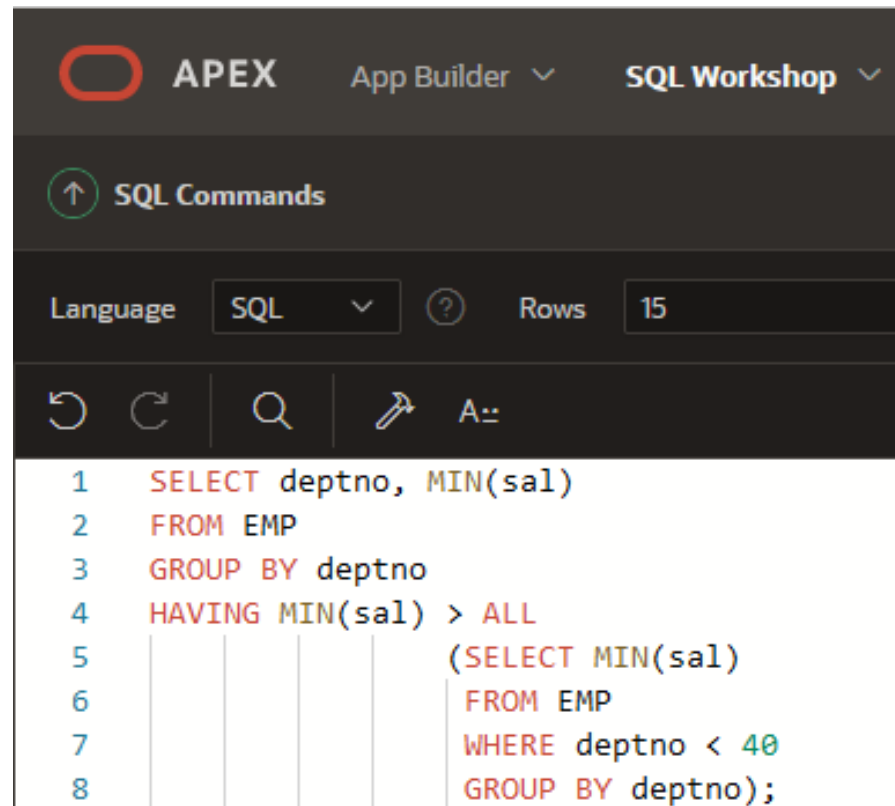
De asemenea, se poate folosi clauza **GROUP BY** intr-o subinterogare

```
SELECT deptno, MIN(sal)
FROM EMP
GROUP BY deptno
HAVING MIN(sal) > ALL
    (SELECT MIN(sal)
     FROM EMP
     WHERE deptno < 40
     GROUP BY deptno);
```

## 6.2. MULTIPLE ROW SUBQUERIES

Clauzele **GROUP BY** si **HAVING**

De asemenea, se poate folosi clauza **GROUP BY** intr-o subinterogare



```
1 SELECT deptno, MIN(sal)
2 FROM EMP
3 GROUP BY deptno
4 HAVING MIN(sal) > ALL
5      (SELECT MIN(sal)
6       FROM EMP
7       WHERE deptno < 40
8       GROUP BY deptno);
```

Results	Explain	Describe	Saved SQL	History
no data found				

## 6.2. MULTIPLE ROW SUBQUERIES

### Aplicatii rezolvate:

- 1) Găsiți numele pentru toți angajații ale căror salarii sunt aceleași cu salariul minim din oricare (**any**) departament.

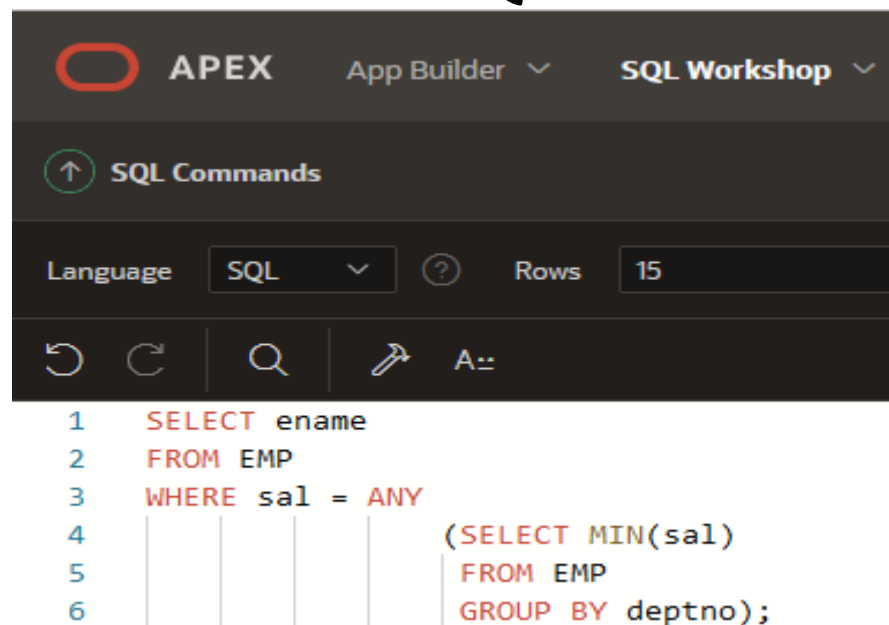
```
SELECT ename  
FROM EMP  
WHERE sal = ANY  
           (SELECT MIN(sal)  
            FROM EMP  
            GROUP BY deptno);
```



## 6.2. MULTIPLE ROW SUBQUERIES

### Aplicatii rezolvate:

- 1) Găsiți numele pentru toți angajații ale căror salarii sunt aceleași cu (**any**) departament. salariul minim din oricare



```
1 SELECT ename
2 FROM EMP
3 WHERE sal = ANY
4         (SELECT MIN(sal)
5          FROM EMP
6          GROUP BY deptno);
```

Results				Explain	Describe	Saved SQL	History
				ENAME			
SMITH							
JAMES							
MILLER							
3 rows returned in 0.01 seconds				Download			

## 6.2. MULTIPLE ROW SUBQUERIES

2) Scopul interogării următoare este de a afișa salariul minim pentru fiecare departament al cărui salariu minim este mai mic decât cel mai mic salariu al angajaților din departamentul 30.

Oricum, subinterogarea nu se execută deoarece are 5 erori.

Găsiți erorile și corectați-le.

**SELECT** deptno

**FROM** EMP

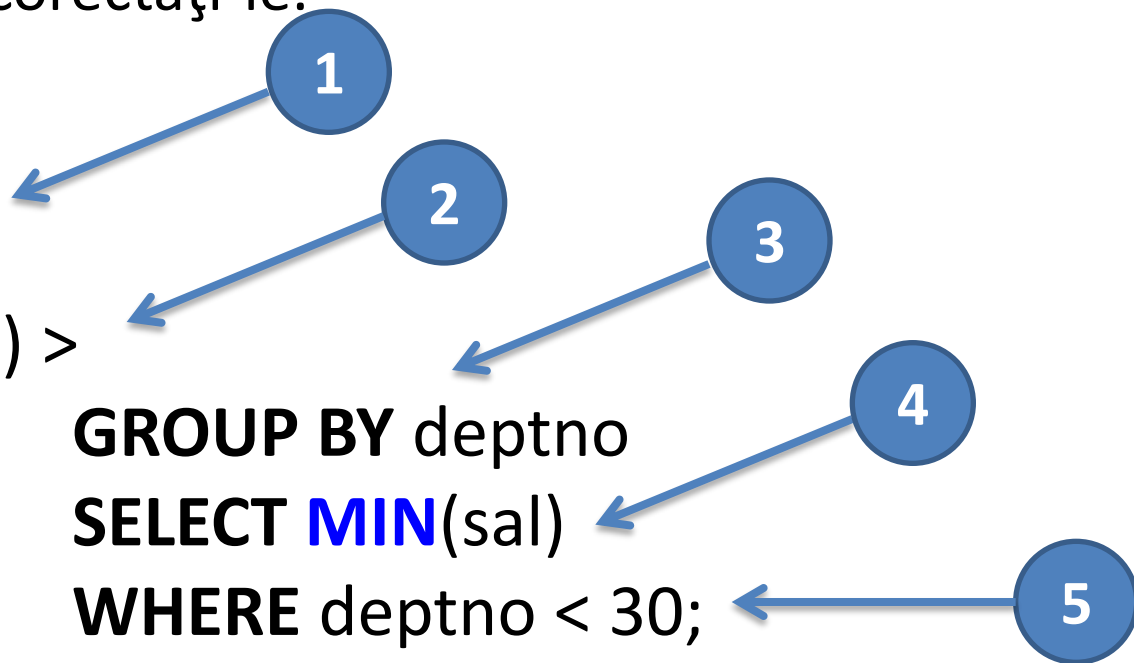
**WHERE** **MIN**(sal)

**HAVING** **MIN**(sal) >

**GROUP BY** deptno

**SELECT** **MIN**(sal)

**WHERE** deptno < 30;



## 6.2. MULTIPLE ROW SUBQUERIES

Soluția corectă este următoarea:

```
SELECT deptno, MIN(sal)
FROM EMP
GROUP BY deptno
HAVING MIN(sal) <
        ( SELECT MIN(sal)
          FROM EMP
          WHERE deptno = 30);
```

## Subcereri multilinie

- Subcererile multilinie returneaza mai mult decat o linie.
- Cu astfel de subinterogari trebuie folositi operatori multilinie care pot prelucra una sau mai multe valori.

Operatorii utilizati sunt:

1. **IN** - egal cu oricare dintre membrii unei liste
2. **ANY/SOME** - compara o valoare cu fiecare (vreo) valoare returnata de subinterogare
3. **ALL** - compara o valoare cu oricare (toate) din valorile returnate de subinterogare

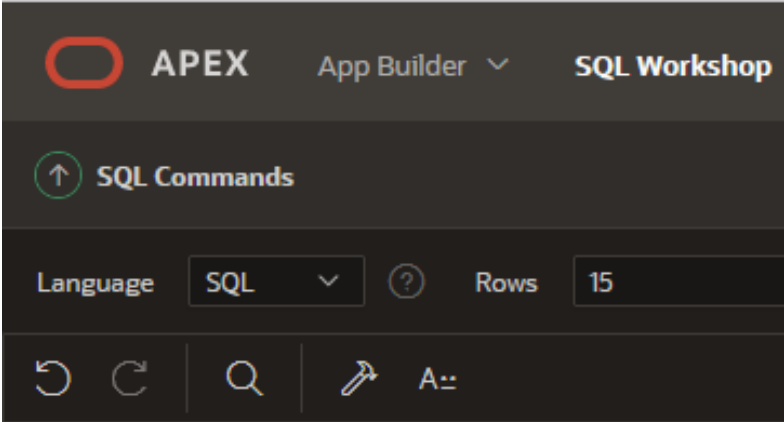
## Exemplu **IN**

Aflati angajatii care au salariul egal cu salariul cel mai mare din fiecare departament

```
SELECT ename, deptno, sal  
FROM EMP  
WHERE sal IN  
      ( SELECT MAX(sal)  
        FROM EMP  
        GROUP BY deptno )
```

subinterogarea ofera salariile maxime din fiecare departament si prin interogarea principala se afla angajatii cu aceste salarii.

Exemplu **IN** - Aflati angajatii care au salariul egal cu salariul cel mai mare din fiecare departament



The screenshot shows the APEX SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', and 'SQL Workshop'. Below it, a 'SQL Commands' section has a 'Language' dropdown set to 'SQL' and a 'Rows' limit of '15'. The main area contains an SQL query:

```
1 SELECT ename, deptno, sal
2 FROM EMP
3 WHERE sal IN
4         ( SELECT MAX(sal)
5           FROM EMP
6           GROUP BY deptno );
7
```

Results			Explain	Describe	Saved SQL	History
ENAME			DEPTNO			SAL
KING			10			5000
BLAKE			30			2850
SCOTT			20			3000
FORD			20			3000

4 rows returned in 0.01 seconds [Download](#)

## Exemplu **ANY**

Aflati angajatii care au salariul mai mare decat vreun angajat al departamentului 20 si nu fac parte din acest departament.

```
SELECT ename, deptno, sal  
FROM EMP  
WHERE sal > ANY  
           ( SELECT sal  
             FROM EMP  
             WHERE deptno = 20 )  
AND deptno <> 20;
```

Exemplu **ANY** - Aflati angajatii care au salariul mai mare decat vreun angajat al departamentului 20 si nu fac parte din acest departament.

```

SQL Commands

Language SQL Rows 15

1 SELECT ename, deptno, sal
2 FROM EMP
3 WHERE sal > ANY
4      ( SELECT sal
5        FROM EMP
6        WHERE deptno = 20 )
7 AND deptno <> 20;

```

Results	Explain	Describe	Saved SQL	History
ENAME	DEPTNO	SAL		
KING	10	5000		
BLAKE	30	2850		
CLARK	10	2450		
ALLEN	30	1600		
TURNER	30	1500		
MILLER	10	1300		
MARTIN	30	1250		
WARD	30	1250		
JAMES	30	950		

9 rows returned in 0.01 seconds Download



Operatorul **ANY** (sinonim operatorului **SOME**) compara o valoare cu fiecare valoare din cele returnate de subinterogare.

Astfel,

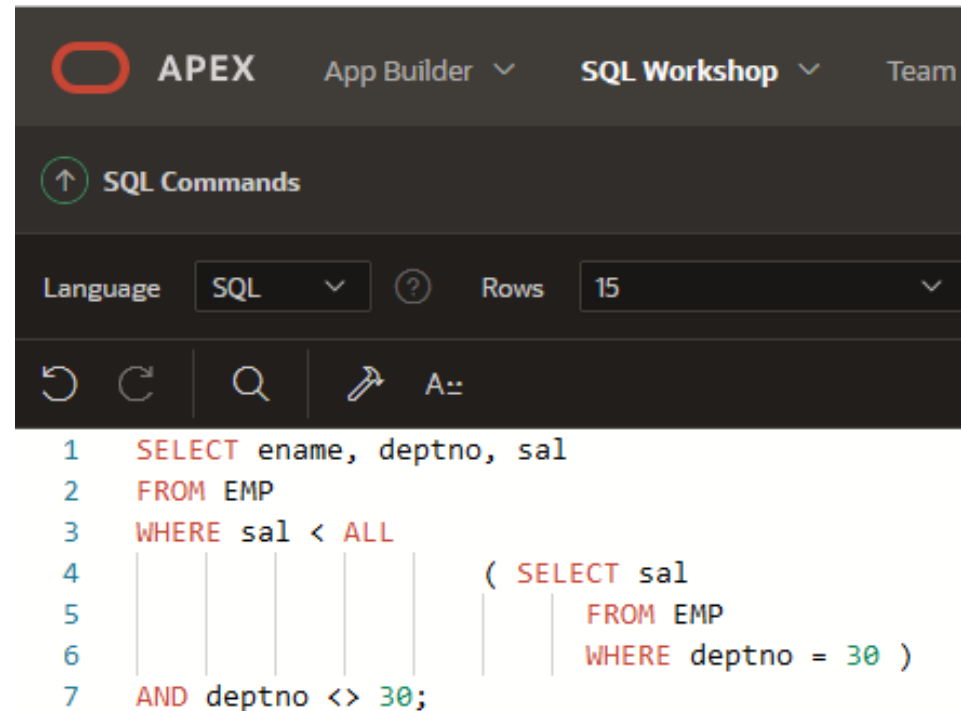
- < ANY** inseamna mai mic decat maximul
- ANY** inseamna mai mare decat minimul
- = ANY** este echivalent cu **IN**

## Exemplu **ALL**

Gasiti angajatii care au salariul mai mic decat oricare (toti) angajatii de la departamentul 30.

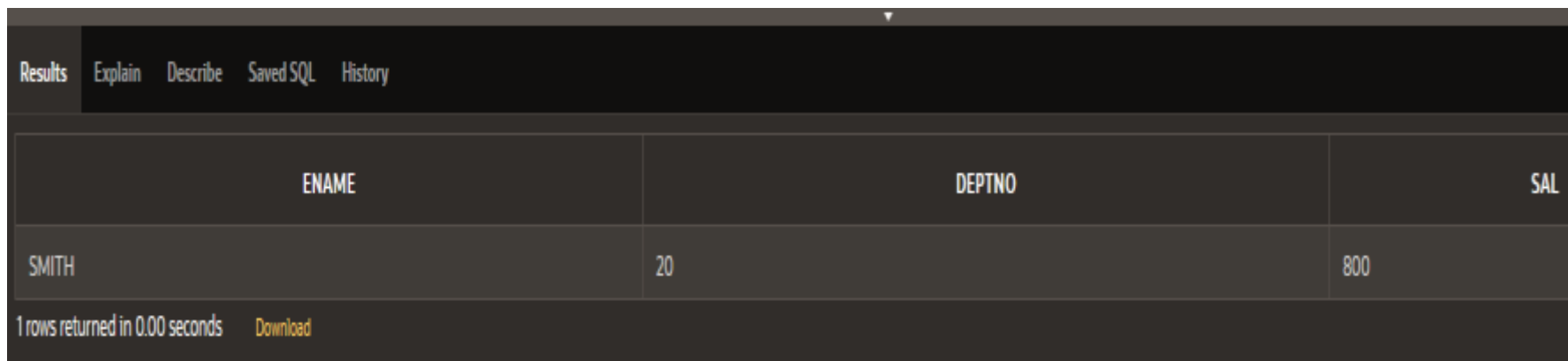
```
SELECT ename, deptno, sal  
FROM EMP  
WHERE sal < ALL  
                ( SELECT sal  
                  FROM EMP  
                  WHERE deptno = 30 )  
AND deptno <> 30;
```

Exemplu **ALL** - Gasiti angajatii care au salariul mai mic decat oricare (toti) angajatii de la departamentul 30.



The screenshot shows the APEX SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', 'SQL Workshop', and 'Team'. Below this is a 'SQL Commands' section with a 'Language' dropdown set to 'SQL' and a 'Rows' limit of '15'. The main area displays a SQL query:

```
1 SELECT ename, deptno, sal
2 FROM EMP
3 WHERE sal < ALL
4      ( SELECT sal
5        FROM EMP
6        WHERE deptno = 30 )
7 AND deptno <> 30;
```



The screenshot shows the 'Results' tab of the APEX SQL Workshop. It displays the output of the SQL query in a table with three columns: ENAME, DEPTNO, and SAL. The table contains one row with the values SMITH, 20, and 800. Below the table, it indicates '1 rows returned in 0.00 seconds' and provides a 'Download' link.

ENAME	DEPTNO	SAL
SMITH	20	800

1 rows returned in 0.00 seconds [Download](#)

- Operatorul **ALL** din interogarea principala compara o valoare cu oricare valoare returnata de subinterogare.

Astfel:

> **ALL** inseamna mai mare decat maximul

< **ALL** inseamna mai mic decat minimul

## Imbricarea subcererilor

Subcererile pot fi folosite si in interiorul altor subinterogari.

### Exemplu

Gasiti numele, functia, data angajarii si salariul angajatilor al caror salariu este superior celui mai mare salariu al vreunei persoane angajate dupa data de 12/09/1982.

```
SELECT ename, job, hiredate, sal
FROM EMP
WHERE sal >
    ( SELECT MAX(sal)
      FROM EMP
      WHERE hiredate IN
          ( SELECT hiredate
            FROM EMP
            WHERE hiredate > '12/09/1982' ) );
```

Numarul maxim de imbricari pentru o subinterogare este de 255.

Numarul maxim de  
imbricari pentru o  
subinterogare este  
de 255.

```

1 SELECT ename, job, hiredate, sal
2 FROM EMP
3 WHERE sal >
4     ( SELECT MAX(sal)
5       FROM EMP
6       WHERE hiredate IN
7         ( SELECT hiredate
8           FROM EMP
9           WHERE hiredate > '12/09/1982' ) );
10

```

ENAME	JOB	HIREDATE	SAL
KING	PRESIDENT	11/17/1981	5000
BLAKE	MANAGER	05/01/1981	2850
CLARK	MANAGER	06/09/1981	2450
JONES	MANAGER	04/02/1981	2975
SCOTT	ANALYST	12/09/1982	3000
FORD	ANALYST	12/03/1981	3000
ALLEN	SALESMAN	02/20/1981	1600
WARD	SALESMAN	02/22/1981	1250
MARTIN	SALESMAN	09/28/1981	1250
TURNER	SALESMAN	09/08/1981	1500
MILLER	CLERK	01/23/1982	1300

11 rows returned in 0.01 seconds [Download](#)

## Subcereri corelate

- O subinterogare corelata este o subinterogare care se executa o data pentru fiecare linie considerata de interogarea principala si care la executie foloseste o valoare dintr-o coloana din interogarea exterioara.
- Ea se poate identifica prin folosirea unei coloane a interogarii exterioare in clauza operatorului interogarii interioare.



## Exemplu

Gasiti angajatii care au un salariu superior salariului mediu al departamentului lor.

```
SELECT ename, sal, deptno  
FROM EMP E  
WHERE sal >  
        ( SELECT AVG(sal)  
          FROM EMP  
          WHERE ( deptno = E.deptno ) )  
ORDER BY deptno;
```

Exemplu - Gasiti  
angajatii care au un  
salariu superior  
salariului mediu al  
departamentului lor.

The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes 'APEX', 'App Builder', 'SQL Workshop', and 'Team Dev'. Below this is a 'SQL Commands' section with a 'Language' dropdown set to 'SQL' and a 'Rows' dropdown set to '15'. The SQL editor contains the following query:

```

1  SELECT ename, sal, deptno
2  FROM EMP E
3  WHERE sal >
4         ( SELECT AVG(sal)
5           FROM EMP
6           WHERE ( deptno = E.deptno ) )
7  ORDER BY deptno;

```

Results			Explain	Describe	Saved SQL	History
ENAME	SAL	DEPTNO				
KING	5000	10				
JONES	2975	20				
SCOTT	3000	20				
FORD	3000	20				
BLAKE	2850	30				
ALLEN	1600	30				

6 rows returned in 0.03 seconds [Download](#)

## Valori de NULL intr-o subinterogare

- In cazul in care subinterogarea returneaza vreuna din valori NULL si interogarea principala are operator NOT IN, atunci interogarea principala nu va returna niciun rand.
- Motivul este ca *o comparatie cu NULL conduce la un rezultat NULL*.

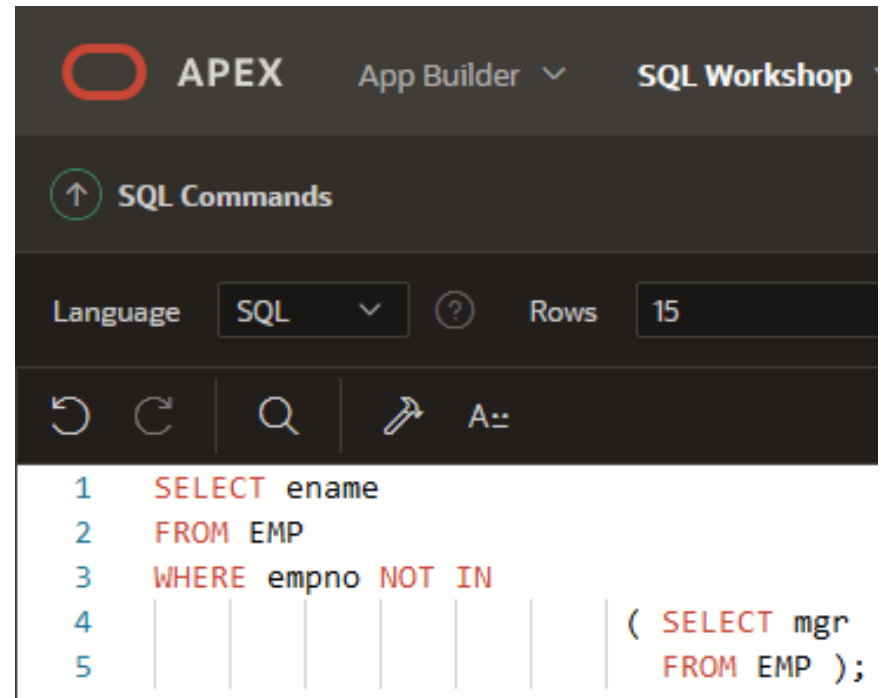
## Exemplu

Gasiti angajatii care nu au subordonati.

```
SELECT ename  
FROM EMP  
WHERE empno NOT IN  
      ( SELECT mgr  
        FROM EMP );
```

## Exemplu

Gasiti angajatii care **nu au** subordonati.



The screenshot shows the APEX SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', and 'SQL Workshop'. Below this is a 'SQL Commands' section with an upward arrow icon. The 'Language' is set to 'SQL' and 'Rows' is set to '15'. The SQL editor contains the following query:

```
1  SELECT ename
2  FROM EMP
3  WHERE empno NOT IN
4      ( SELECT mgr
5        FROM EMP );
```

Results	Explain	Describe	Saved SQL	History
no data found				

- Astfel ori de cate ori valoarea **NULL** face parte din raspunsurile subcererii nu trebuie folosit operatorul **NOT IN**.
- De fapt operatorul **NOT IN** este echivalent cu **<> ALL**.
- Returnarea de valori NULL de catre subinterogare nu prezinta nici o problema in cazul operatorului **IN** in interogarea principala (in echivalent cu **= ALL**).

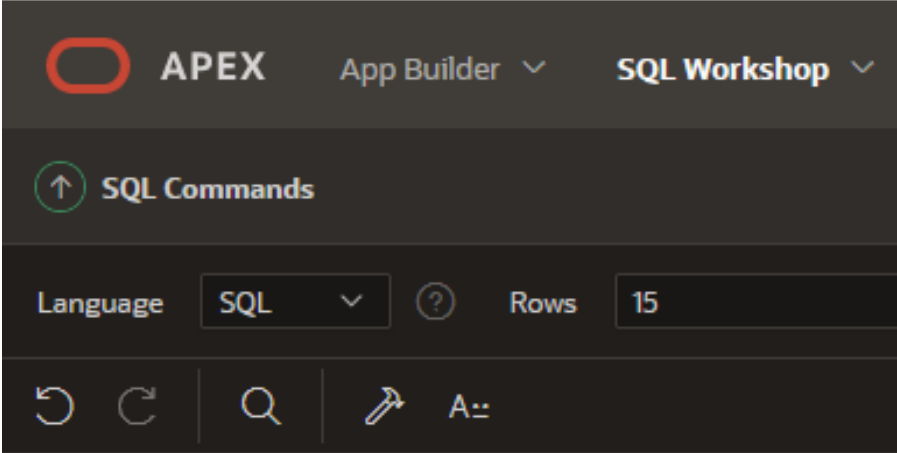
## Exemplu

Gasiti angajatii care **au** subordonati.

```
SELECT ename  
FROM EMP  
WHERE empno IN  
      ( SELECT mgr  
        FROM EMP );
```

## Exemplu

Gasiti angajatii care au subordonati.



The screenshot shows the APEX SQL Workshop interface. At the top, there's a navigation bar with 'APEX', 'App Builder', and 'SQL Workshop'. Below this is a 'SQL Commands' section with a 'Language' dropdown set to 'SQL' and a 'Rows' limit of '15'. The SQL editor contains the following query:

```
1 SELECT ename
2 FROM EMP
3 WHERE empno IN
4      ( SELECT mgr
5        FROM EMP );
```



The screenshot shows the 'Results' tab of the APEX SQL Workshop. The table displays the results of the SQL query, showing the names of employees who have subordinates. The table has a single column labeled 'ENAME'.

ENAME
KING
BLAKE
CLARK
JONES
SCOTT
FORD

6 rows returned in 0.03 seconds [Download](#)

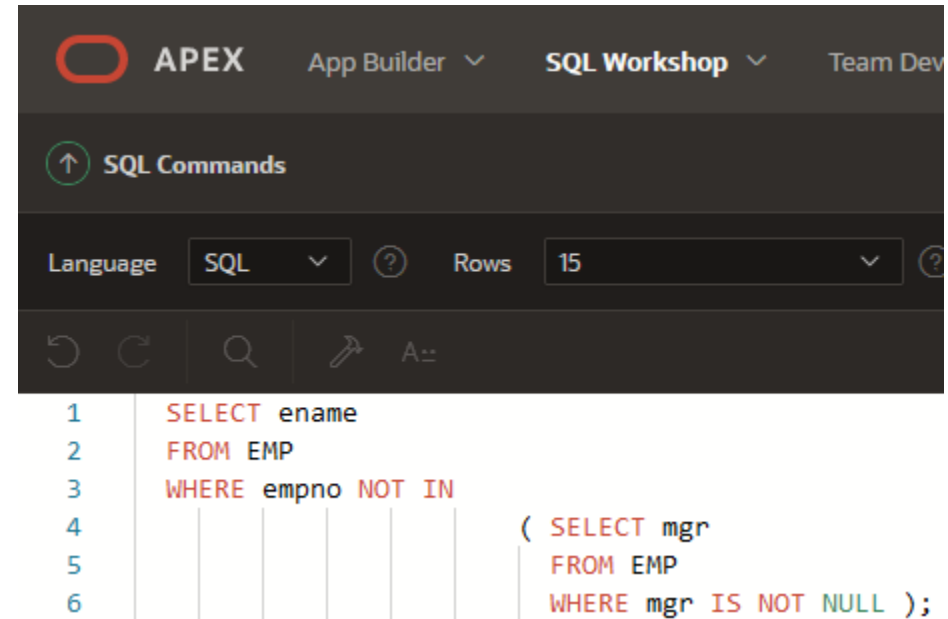


- In cazul utilizarii operatorului **NOT IN** in interogarea principala trebuie avut grija sa se excluda valorile **NULL** din raspunsurile subcererii.

## Exemplu

Gasiti angajatii care **nu au** subordonati.

```
SELECT ename
FROM EMP
WHERE empno NOT IN
      ( SELECT mgr
        FROM EMP
        WHERE mgr IS NOT NULL );
```



The screenshot shows the APEX SQL Workshop interface. The top navigation bar includes the APEX logo, 'App Builder', 'SQL Workshop', and 'Team Dev'. Below this is a 'SQL Commands' section with a search icon and a 'Language' dropdown set to 'SQL'. The 'Rows' dropdown is set to '15'. The SQL editor shows the following query:

```
1 SELECT ename
2 FROM EMP
3 WHERE empno NOT IN
4      ( SELECT mgr
5        FROM EMP
6        WHERE mgr IS NOT NULL );
```

Results	Explain	Describe	Saved SQL	History
ENAME				
SMITH				
ALLEN				
WARD				
MARTIN				
TURNER				
ADAMS				
JAMES				
MILLER				
8 rows returned in 0.01 seconds				
<a href="#">Download</a>				

## Sfaturi în utilizarea subinterogărilor

1. Includerea subinterogărilor în paranteze
2. Plasarea subinterogărilor în partea dreapta a operatorului de comparare
3. A nu se adauga clauza **ORDER BY** într-o subinterogare
4. Folosirea operatorilor single-row în subinterogari single-row
5. Folosirea operatorilor multiple-row în subinterogari multiple-row

# Concluzii

1. O subinterogare este o instructiune **SELECT** inclusa într-o clauza a altei instructiuni **SQL**.
2. Subinterogările sunt folosite atunci când interogarea se bazează pe criterii necunoscute.
3. Subinterogările au următoarele caracteristici:
  - a) Pot transmite un rand de date instructiunii principale care contine un **operator single-row**, precum: **=**, **<>**, **>**, **>=**, **<** sau **<=**;
  - b) Pot transmite rânduri multiple de date instructiunii principale care contine un **operator multiple-row**, precum: **IN**, **ANY** sau **ALL**;
  - c) Sunt primele procesate de către server-ul **Oracle**, iar clauzele **WHERE** și **HAVING** folosesc rezultatele;
  - d) Pot contine functii de grup.

# Întrebări?