

Baze de date

Limbajul SQL



THE **INFORMATION** COMPANY

Curs 9

Limbajul SQL

Limbajul SQL

Vederi (Views)

O vedere este o tabelă logică care extrage date dintr-o tabelă propriu-zisă sau dintr-o altă vedere.

O vedere nu are date proprii, ci este ca o fereastră prin care datele din tabele pot fi actualizate sau vizualizate.

O vedere poate să conțină toate datele dintr-o tabelă (sau mai multe) sau anumite coloane.

Vederile se folosesc pentru că oferă:

1. Acces restricționat la date (vederile pot selecționa anumite coloane dintr-o tabelă)
2. Posibilitatea lucrului cu interogări complexe
3. Independența datelor
4. Posibilitatea prezentării de date identice în vederi diferite

Avantajele utilizării vederilor

- Vederile pot fi folosite în cereri **SQL** simple când de fapt ele întorc rezultatul unor cereri **SQL** complexe.
 - Deci putem utiliza vederile în interogarea mai multor tabele fără a ști ce este un join de tabele.
- O vedere poate fi folosită pentru a obține date din mai multe tabele.
- Vederile oferă drepturi de acces pentru grupuri de utilizatori.

Clasificare

Vederile se clasifică în:

1. vederi **simple**
2. vederi **complexe**

Pentru început prezentăm **vederile simple**.

Crearea vederilor

Vederile simple sunt acele vederi care:

1. Interoghează o singură tabelă
2. Nu conțin funcții
3. Nu conțin grupuri de date
4. Oferă posibilitatea efectuării de operații **DML**
(**INSERT, UPDATE, DELETE**)

NOTĂ:

O vedere are un nume și este generată de o cerere **SQL**.

Sintaxa

```
CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW  
view    [ (alias [, alias]...) ]  
    AS subquery  
[WITH CHECK OPTION [CONSTRAINT  
constraint] ]  
[WITH READ ONLY [CONSTRAINT constraint] ];
```

Semnificația argumentelor sintaxei:

- **OR REPLACE**: determină înlocuirea vederii dacă aceasta există deja
- **FORCE**: crearea vederii indiferent dacă tabela din care provin datele nu există
- **NOFORCE**: crearea vederii numai dacă tabela există (valoarea implicită)
- **view**: numele vederii
- **alias**: specifică numele expresiilor selectate de către cererea din vedere (numele coloanelor selectate). Numărul alias-urilor trebuie să fie egal cu numărul coloanelor selectate.

- **subquery**: cererea SELECT care generează vederea
- **WITH CHECK OPTION**: numai rândurile accesibile vederii pot fi actualizate
- **constraint**: numele constrângerii asociate cu WITH CHECK OPTION
- **WITH READ ONLY**: asigură că nici o operație **DML (INSERT, UPDATE, DELETE)** nu va fi executată prin vedere.

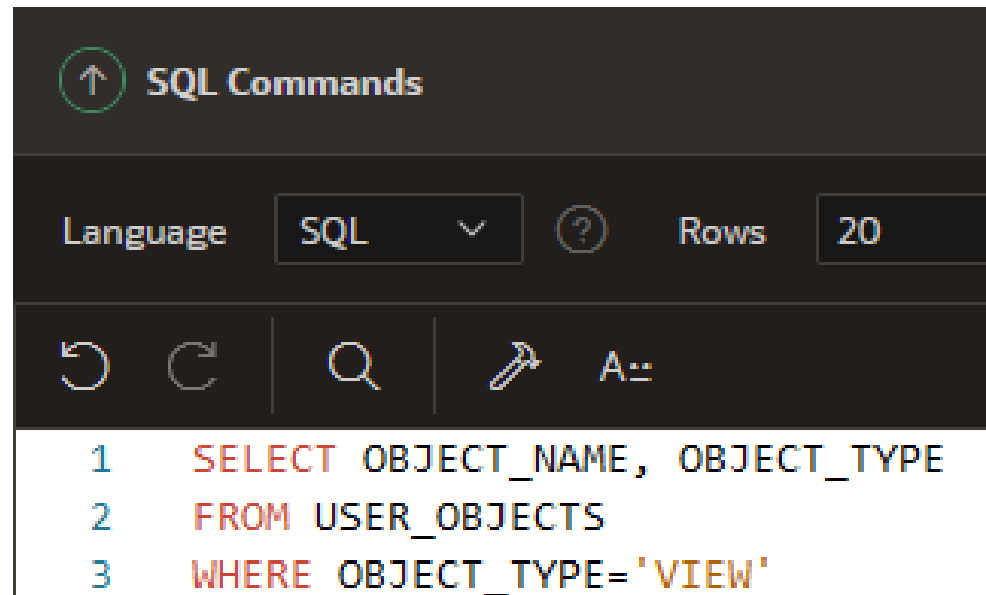
Exemplu

Să cream o vedere **emp20** care va conține persoanele din departamentul 20.

Mai întâi trebuie să ne asigurăm că vederea **emp20** nu există deja.

```
SELECT OBJECT_NAME, OBJECT_TYPE  
FROM USER_OBJECTS  
WHERE OBJECT_TYPE='VIEW'
```

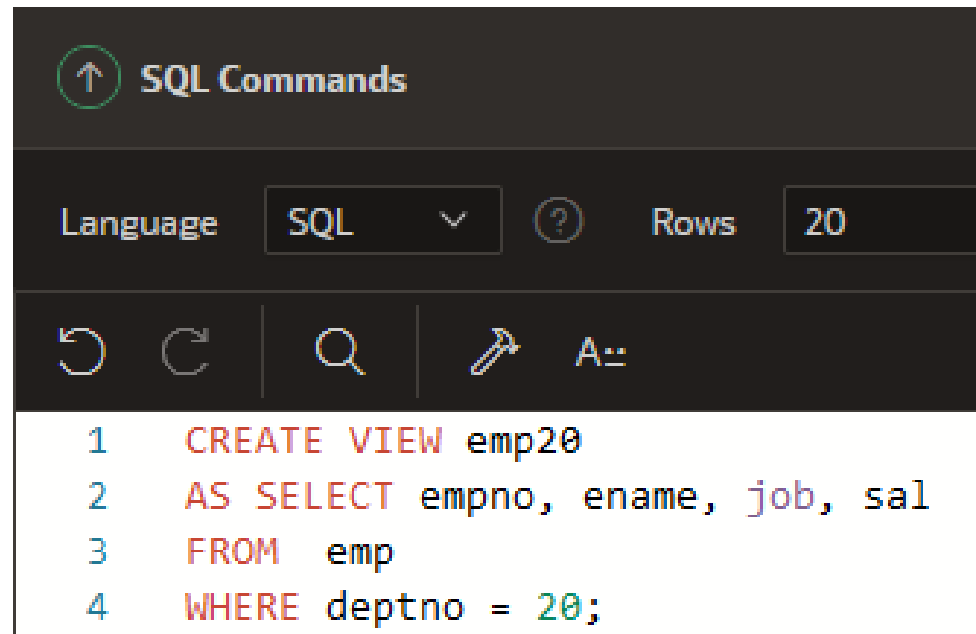
Această vedere nu există deci putem să o cream.



The screenshot shows a SQL editor window titled "SQL Commands". It has a "Language" dropdown set to "SQL" and a "Rows" limit of "20". Below the toolbar, the following SQL command is entered:

```
1 SELECT OBJECT_NAME, OBJECT_TYPE  
2 FROM USER_OBJECTS  
3 WHERE OBJECT_TYPE='VIEW'
```

```
CREATE VIEW emp20  
AS SELECT empno, ename, job, sal  
FROM emp  
WHERE deptno = 20;
```



The screenshot shows a SQL command interface with a dark theme. At the top, there is a tab labeled "SQL Commands" with an upward arrow icon. Below the tab, there is a "Language" dropdown menu set to "SQL", a help icon (question mark), and a "Rows" dropdown menu set to "20". Below these controls is a toolbar with icons for undo, redo, search, and a save icon. The main area displays the SQL command being executed, line by line:

```
1 CREATE VIEW emp20  
2 AS SELECT empno, ename, job, sal  
3 FROM emp  
4 WHERE deptno = 20;
```

Pentru a vedea faptul că vederea a fost creată într-adevăr folosim aceeași sintaxă ca mai sus, deci vom interoga tabela de obiecte **USER_OBJECTS**.

```
SELECT OBJECT_NAME, OBJECT_TYPE  
FROM USER_OBJECTS  
WHERE OBJECT_TYPE = 'VIEW'
```

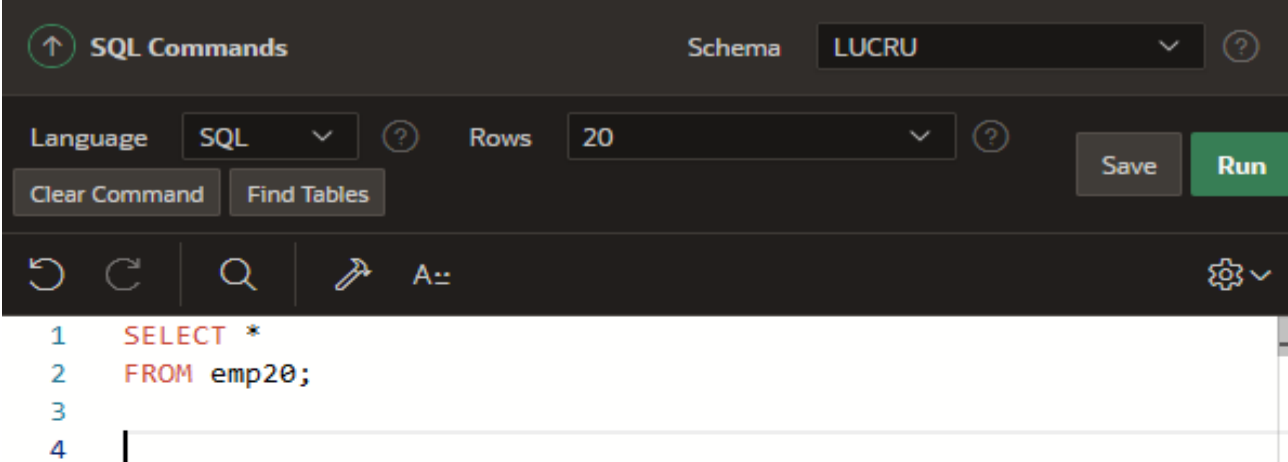
The screenshot shows a SQL IDE interface. At the top, the 'SQL Commands' tab is active, displaying the query: `SELECT OBJECT_NAME, OBJECT_TYPE FROM USER_OBJECTS WHERE OBJECT_TYPE = 'VIEW'`. The 'Schema' dropdown is set to 'LUCRU'. Below the query editor, the 'Results' tab is selected, showing a table with two columns: 'OBJECT_NAME' and 'OBJECT_TYPE'. The table contains 8 rows of data, all with 'VIEW' as the object type. At the bottom, a status bar indicates '8 rows returned in 0.05 seconds' and provides a 'Download' link.

OBJECT_NAME	OBJECT_TYPE
CUSTOMER_ORDER_PRODUCTS	VIEW
EBA_COUNTRIES_V	VIEW
EMP20	VIEW
EMP_DEPT_V	VIEW
PRODUCT_ORDERS	VIEW
PRODUCT_REVIEWS	VIEW
STORE_ORDERS	VIEW
STORE_ORDERS_STATUS	VIEW

8 rows returned in 0.05 seconds [Download](#)

Pentru a vizualiza conținutul vederii folosim cererea **SQL**
SELECT.

SELECT *
FROM emp20;



The screenshot shows a SQL IDE interface. At the top, there's a 'SQL Commands' tab with a 'Schema' dropdown set to 'LUCRU'. Below this, there are controls for 'Language' (set to 'SQL'), 'Rows' (set to '20'), and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. The main editor area contains the following SQL query:

```
1 SELECT *
2 FROM emp20;
3
4
```

Below the editor, there's a 'Results' tab with sub-tabs for 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying a table with 4 columns: 'EMPNO', 'ENAME', 'JOB', and 'SAL'. The table contains 6 rows of data. At the bottom, it states '6 rows returned in 0.01 seconds' and has a 'Download' button.

EMPNO	ENAME	JOB	SAL
7566	JONES	MANAGER	2975
7788	SCOTT	ANALYST	3000
7902	FORD	ANALYST	3000
7369	SMITH	CLERK	800
7876	ADAMS	CLERK	1100
9999	IONESCU	SALESMAN	1000

Reguli pentru crearea vederilor

- Subcererea care definește vederea poate conține **cereri SELECT complexe** incluzând join-uri, grupuri sau alte subcereri.
- Subcererea principală a vederii nu trebuie să conțină clauza **ORDER BY**. Aceasta se utilizează când se extrag date din vedere.
- Dacă nu se specifică un nume pentru constrângerea **WITH CHECK OPTION**, sistemul atribuie în mod implicit un nume
- Cu ajutorul opțiunii **OR REPLACE** vederea este alterată fără a o distruge, a o recrea și a reacorda drepturi asupra ei.

Crearea unei vederi cu alias-uri de coloană

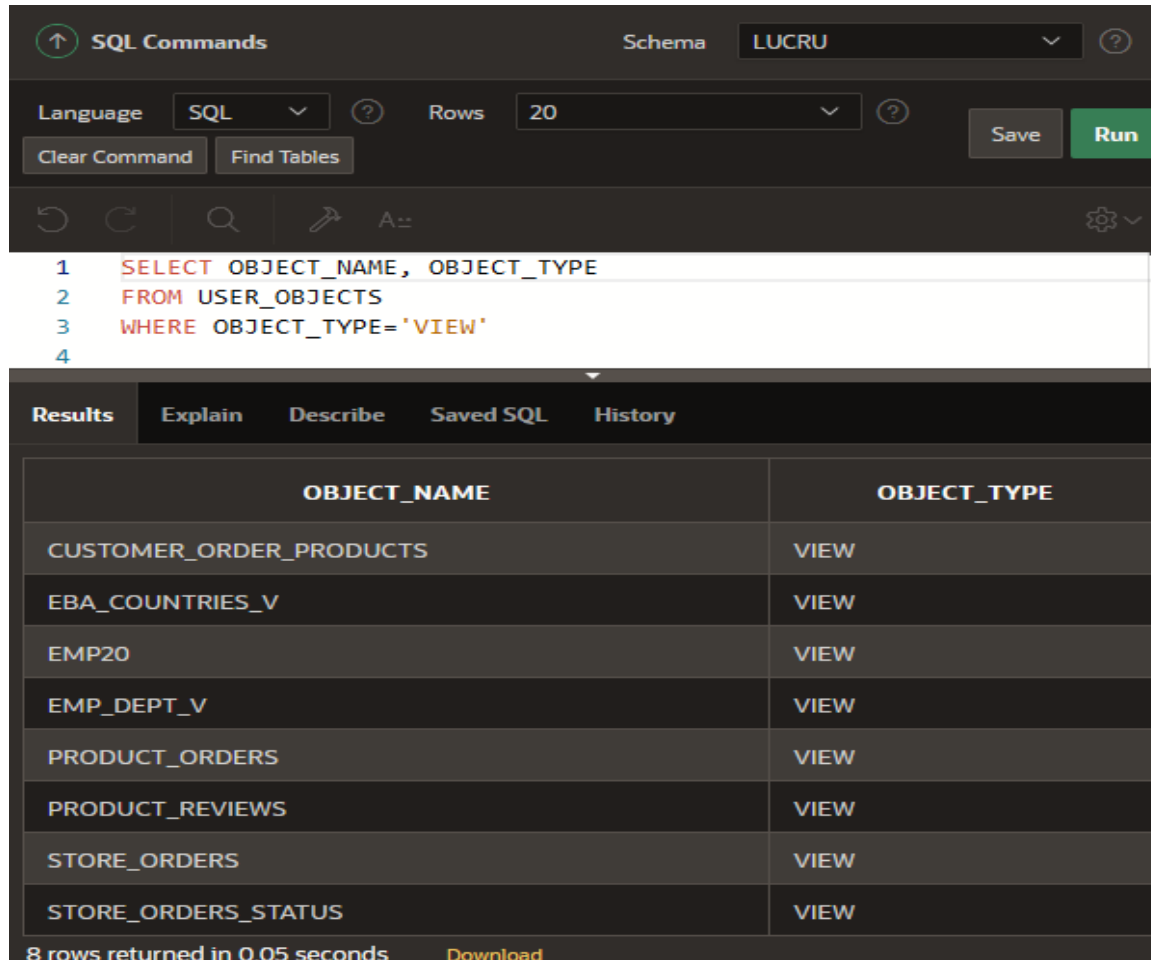
Exemplu

Să cream o vedere **emp10** care va conține persoanele din departamentul 10.

Mai întâi trebuie să ne asigurăm că vederea **emp10** nu există deja.

```
SELECT OBJECT_NAME, OBJECT_TYPE  
FROM USER_OBJECTS  
WHERE OBJECT_TYPE='VIEW';
```

Această vedere nu există deci putem să o cream.



The screenshot shows a SQL IDE interface. At the top, the 'SQL Commands' tab is active, with the 'Schema' dropdown set to 'LUCRU'. The 'Language' is set to 'SQL' and 'Rows' is set to '20'. The query is entered in the command area and is as follows:

```
1 SELECT OBJECT_NAME, OBJECT_TYPE  
2 FROM USER_OBJECTS  
3 WHERE OBJECT_TYPE='VIEW'  
4
```

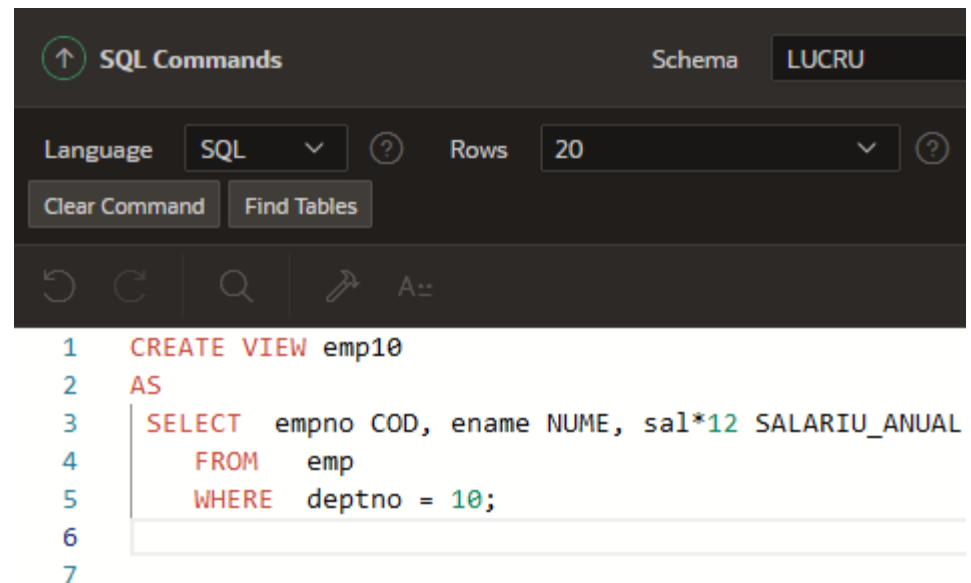
Below the command area, the 'Results' tab is active, displaying the results of the query in a table format. The table has two columns: 'OBJECT_NAME' and 'OBJECT_TYPE'. The results show 8 rows of data, all of which are 'VIEW' objects.

OBJECT_NAME	OBJECT_TYPE
CUSTOMER_ORDER_PRODUCTS	VIEW
EBA_COUNTRIES_V	VIEW
EMP20	VIEW
EMP_DEPT_V	VIEW
PRODUCT_ORDERS	VIEW
PRODUCT_REVIEWS	VIEW
STORE_ORDERS	VIEW
STORE_ORDERS_STATUS	VIEW

At the bottom of the results pane, it states '8 rows returned in 0.05 seconds' and provides a 'Download' link.

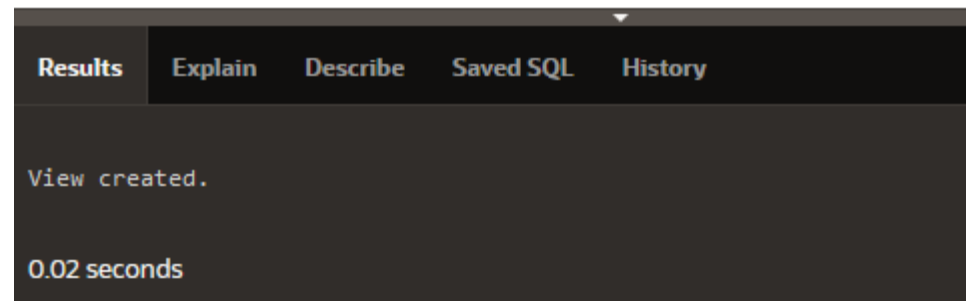
```
CREATE VIEW emp10  
AS  
SELECT empno COD, ename NUME,  
       sal*12 SALARIU_ANUAL  
FROM emp  
WHERE deptno = 10;
```

Alias-uri de
coloane



The screenshot shows a SQL editor window titled 'SQL Commands' with the 'Schema' set to 'LUCRU'. The 'Language' is set to 'SQL' and 'Rows' is set to '20'. The SQL command entered is:

```
1 CREATE VIEW emp10  
2 AS  
3 SELECT empno COD, ename NUME, sal*12 SALARIU_ANUAL  
4 FROM emp  
5 WHERE deptno = 10;  
6  
7
```



The screenshot shows the 'Results' tab of the SQL editor. The message 'View created.' is displayed, along with the execution time '0.02 seconds'.

Pentru a verifica că vederea a fost creată utilizăm comanda de interogare a tabelului **USER_OBJECTS**.

```
SELECT OBJECT_NAME, OBJECT_TYPE  
FROM USER_OBJECTS  
WHERE OBJECT_TYPE='VIEW';
```

The screenshot shows a SQL IDE interface. At the top, the 'SQL Commands' tab is active, with the schema set to 'LUCRU'. The language is 'SQL' and the number of rows to display is '20'. The query entered is:

```
1 SELECT OBJECT_NAME, OBJECT_TYPE  
2 FROM USER_OBJECTS  
3 WHERE OBJECT_TYPE='VIEW';
```

Below the query editor, the 'Results' tab is selected, showing a table with two columns: 'OBJECT_NAME' and 'OBJECT_TYPE'. The table contains 9 rows of data, all of which are views. At the bottom, it indicates '9 rows returned in 0.00 seconds' and provides a 'Download' link.

OBJECT_NAME	OBJECT_TYPE
CUSTOMER_ORDER_PRODUCTS	VIEW
EBA_COUNTRIES_V	VIEW
EMP10	VIEW
EMP20	VIEW
EMP_DEPT_V	VIEW
PRODUCT_ORDERS	VIEW
PRODUCT_REVIEWS	VIEW
STORE_ORDERS	VIEW
STORE_ORDERS_STATUS	VIEW

Pentru a vizualiza conținutul vederii folosim cererea SQL **SELECT**.

Atenție: trebuiesc selectate alias-urile de coloană.

Exercițiu:

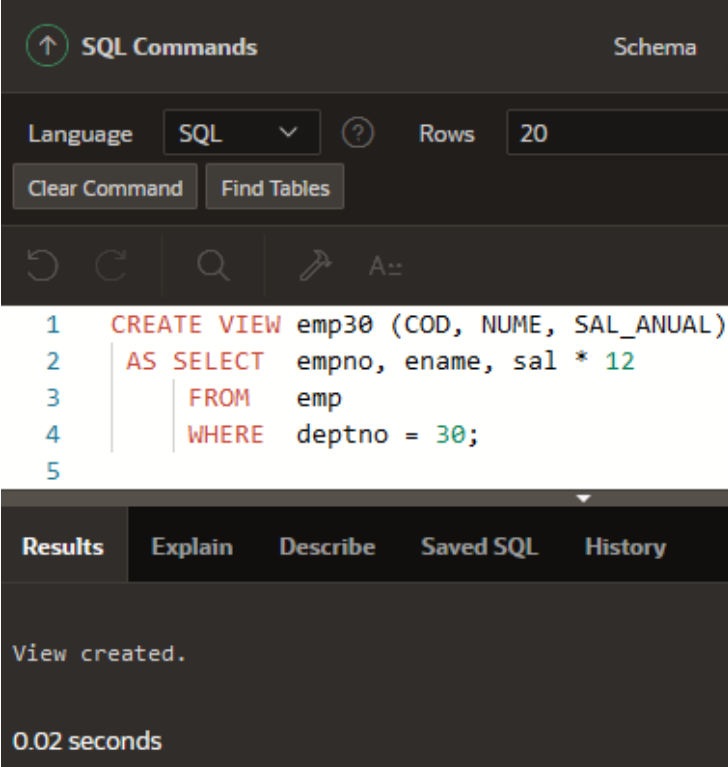
.....

O alternativă la alias-urile utilizate în cererea SQL sunt **alias-urile plasate după CREATE, înaintea cererii SELECT.**

De data aceasta vom crea o vedere cu persoanele din departamentul 30.

Plasarea alias-urilor de coloane
inaintea interogarii

```
CREATE VIEW emp30 (COD, NUME, SAL_ANUAL)
AS SELECT empno, ename, sal * 12
FROM emp
WHERE deptno = 30;
```



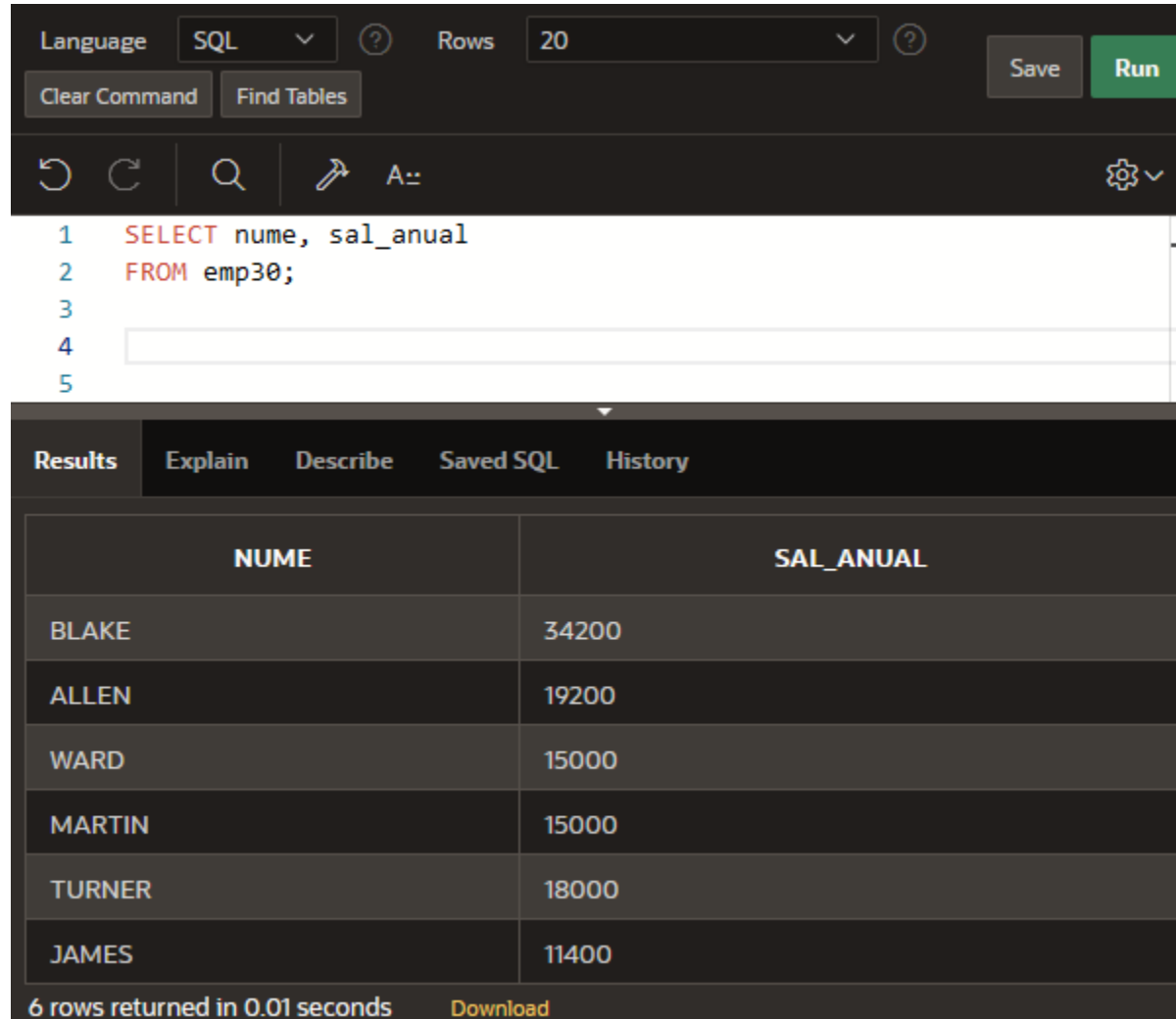
The screenshot shows a SQL IDE interface. The top panel, titled 'SQL Commands', displays the following SQL statement:

```
1 CREATE VIEW emp30 (COD, NUME, SAL_ANUAL)
2 AS SELECT empno, ename, sal * 12
3 FROM emp
4 WHERE deptno = 30;
5
```

The bottom panel shows the 'Results' tab with the message 'View created.' and the execution time '0.02 seconds'.

Exercitiu

Vizualizați numele și salariul anual al persoanelor din vederea **emp30**.



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '20', and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. Below the toolbar is a query editor with the following SQL code:

```
1 SELECT nume, sal_anual
2 FROM emp30;
3
4
5
```

Below the query editor is a 'Results' tab with sub-tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, displaying a table with two columns: 'NUME' and 'SAL_ANUAL'. The table contains six rows of data:

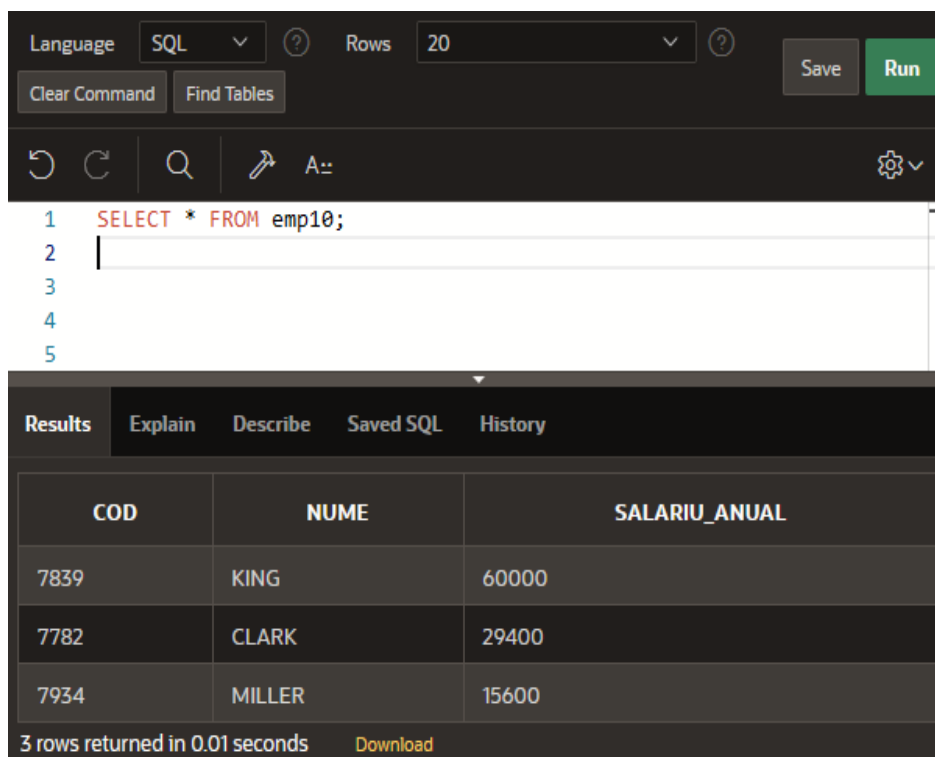
NUME	SAL_ANUAL
BLAKE	34200
ALLEN	19200
WARD	15000
MARTIN	15000
TURNER	18000
JAMES	11400

At the bottom of the results section, it says '6 rows returned in 0.01 seconds' and there is a 'Download' button.

Extragere de date dintr-o vedere

- Extragerea de date se face exact ca în cazul unei tabele.
- Se pot selecta anumite rânduri sau toată vederea.

SELECT * FROM emp10;



The screenshot shows a SQL IDE interface. At the top, there's a toolbar with 'Language' set to 'SQL', 'Rows' set to '20', and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. Below the toolbar is a command input area with the query 'SELECT * FROM emp10;'. The results are displayed in a table with three columns: 'COD', 'NUME', and 'SALARIU_ANUAL'. The table contains three rows of data. At the bottom, it says '3 rows returned in 0.01 seconds' and has a 'Download' button.

COD	NUME	SALARIU_ANUAL
7839	KING	60000
7782	CLARK	29400
7934	MILLER	15600

Interogarea unei vederi

Interogarea unei vederi de către serverul **ORACLE** se face în felul următor:

1. Extragere definiție vedere din tabela dicționarului de date **USER_VIEWS**
2. Verificare drepturi acces pentru tabela vederii
3. Convertire interogare vedere într-o interogare echivalentă a tabelii vederii.

Deci datele sunt extrase, inserate din tabele propriu-zise.

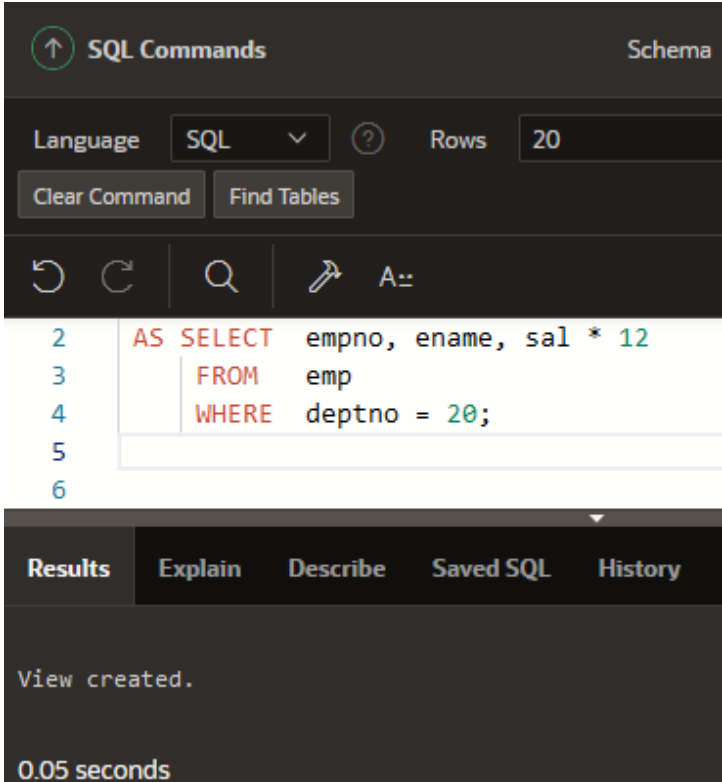
Modificarea unei vederi

Până acum am verificat dacă vederea există sau nu și numai în cazul în care nu exista o cream.

Avem posibilitatea de a crea o vedere chiar dacă ea există, înlocuind-o pe cea veche cu o nouă versiune.

În acest scop se folosește clauza **OR REPLACE**.
Vom înlocui vederea emp20 cu alta.

```
CREATE OR REPLACE VIEW emp20 (COD, NUME,  
    SAL_ANUAL)  
AS SELECT empno, ename, sal * 12  
FROM emp  
WHERE deptno = 20;
```



The screenshot shows a SQL IDE interface. At the top, there's a 'SQL Commands' tab. Below it, the 'Language' is set to 'SQL' and 'Rows' is set to '20'. There are buttons for 'Clear Command' and 'Find Tables'. The main area shows the SQL command being executed, with line numbers 2 through 6 on the left. The command is: `AS SELECT empno, ename, sal * 12 FROM emp WHERE deptno = 20;`. Below the command area, there's a 'Results' tab selected, showing the message 'View created.' and the execution time '0.05 seconds'.

```
2 AS SELECT empno, ename, sal * 12
3 FROM emp
4 WHERE deptno = 20;
5
6
```

Results Explain Describe Saved SQL History

View created.

0.05 seconds

Exercițiu

Vizualizați conținutul vederii cu o cerere **SELECT**.

The screenshot shows a SQL IDE interface. At the top, there's a 'SQL Commands' tab with a 'Schema' dropdown set to 'LUCRU'. Below this, there's a 'Language' dropdown set to 'SQL', a 'Rows' limit of '20', and buttons for 'Clear Command', 'Find Tables', 'Save', and 'Run'. The main query editor contains the following SQL command:

```
1 SELECT * FROM emp20;  
2  
3  
4  
5
```

Below the editor, the 'Results' tab is active, displaying a table with 6 rows. The table has columns 'COD', 'NUME', and 'SAL_ANUAL'. The data is as follows:

COD	NUME	SAL_ANUAL
7566	JONES	35700
7788	SCOTT	36000
7902	FORD	36000
7369	SMITH	9600
7876	ADAMS	13200
9999	IONESCU	12000

At the bottom of the results section, it states '6 rows returned in 0.02 seconds' and provides a 'Download' link.

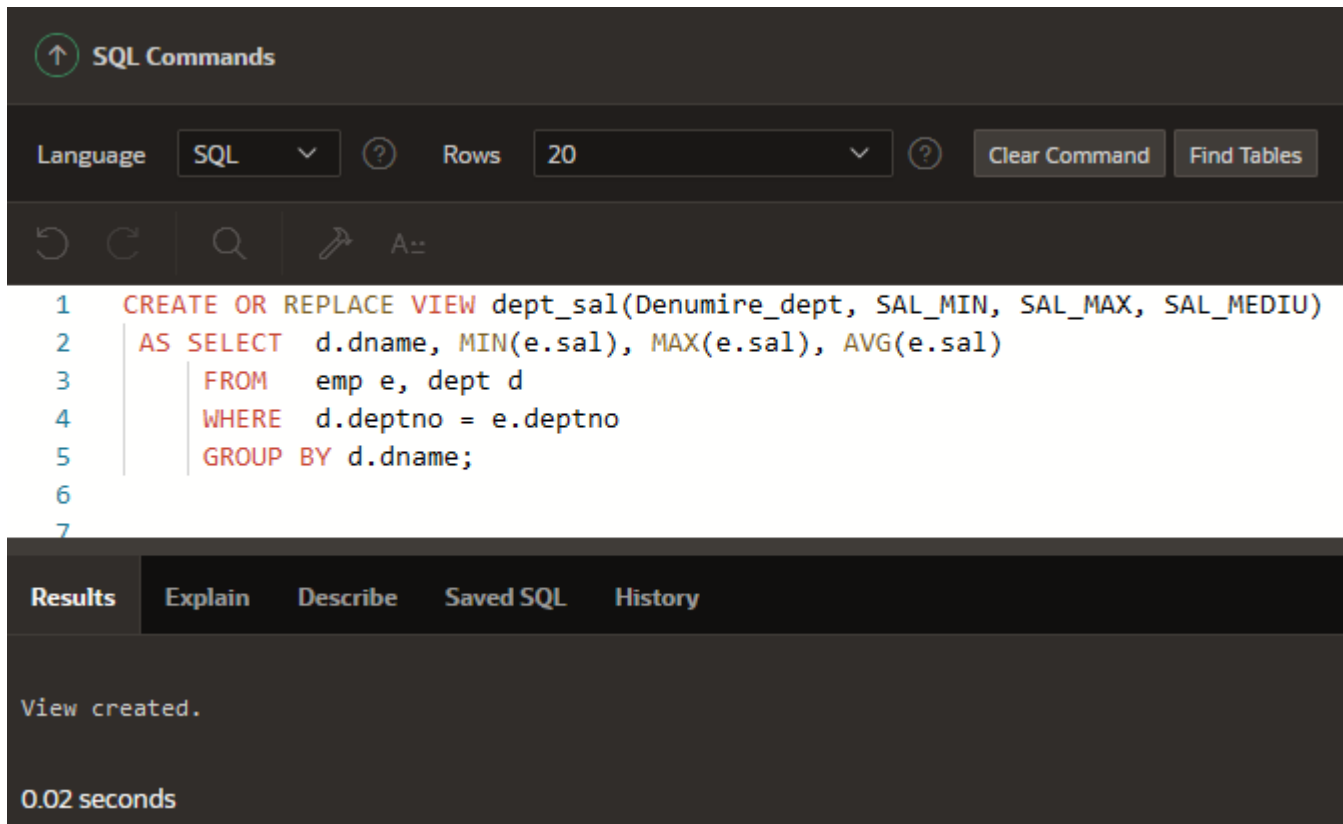
Crearea unei vederi complexe

Vom utiliza funcții de grup pentru a afișa o vedere complexă asupra a 2 tabele.

```
CREATE OR REPLACE VIEW dept_sal(Denumire_dept,  
    SAL_MIN, SAL_MAX, SAL_MEDIU)  
AS SELECT  d.dname, MIN(e.sal), MAX(e.sal),  
    AVG(e.sal)  
FROM    emp e, dept d  
WHERE   d.deptno = e.deptno  
GROUP BY d.dname;
```

Crearea unei vederi complexe

Vom utiliza funcții de grup pentru a afișa o vedere complexă asupra a 2 tabele.



The screenshot shows a SQL IDE interface. At the top, there's a 'SQL Commands' section with a language dropdown set to 'SQL', a 'Rows' limit of 20, and buttons for 'Clear Command' and 'Find Tables'. Below this is a toolbar with icons for undo, redo, search, and a keyboard shortcut 'A++'. The main area contains a SQL command to create or replace a view named 'dept_sal'. The command uses aggregate functions (MIN, MAX, AVG) to calculate salary statistics grouped by department. Below the command, there's a 'Results' tab with sub-tabs for 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing the message 'View created.' and the execution time '0.02 seconds'.

```
1 CREATE OR REPLACE VIEW dept_sal(Denumire_dept, SAL_MIN, SAL_MAX, SAL_MEDIU)
2 AS SELECT d.dname, MIN(e.sal), MAX(e.sal), AVG(e.sal)
3 FROM emp e, dept d
4 WHERE d.deptno = e.deptno
5 GROUP BY d.dname;
6
7
```

Results Explain Describe Saved SQL History

View created.

0.02 seconds

NOTĂ:

- Când o coloană a vederii este o funcție sau o expresie trebuie folosit alias de coloană.
- Conținutul vederii este afișat în același mod ca până acum.

Exercițiu: Afișați conținutul vederii `dept_sal`.

SQL Commands

SchemaLUCRU?

LanguageSQL?

Rows20?

Clear CommandFind TablesSaveRun

↶↷🔍🔧A::⚙️

1SELECT *

2FROM dept_sal;

3

4

5

ResultsExplainDescribeSaved SQLHistory

DENUMIRE_DEPT	SAL_MIN	SAL_MAX	SAL_MEDIU
RESEARCH	800	3000	1979.166666666666666666666666666667
SALES	950	2850	1566.666666666666666666666666666667
ACCOUNTING	1300	5000	2916.666666666666666666666666666667

3 rows returned in 0.03 secondsDownload

Efectuarea operațiilor DML asupra vederilor

Se pot efectua operații **DML** asupra vederilor, dar numai în anumite condiții.

a) *Nu se poate șterge un rând dintr-o vedere* care conține:

1. funcții de grup
2. GROUP BY
3. DISTINCT
4. ROWNUM

- b) De asemenea *nu se pot modifica date printr-o vedere* dacă aceasta verifică condițiile de mai sus și dacă există coloane definite prin expresii cum ar fi SALARIU*12.
- c) *Adăugarea de date printr-o vedere este imposibilă* dacă vederea conține elementele enumerate mai sus și dacă există coloane **NOT NULL** în tabela neselectate de către vedere.

Prin adăugare de date se modifică de fapt tabela propriu-zisă deci în coloanele neselectate se va adăuga **NULL** ceea ce încalcă constrângerea.

Exemplu - Crearea vederii **dept_sal** utilizând clauza **OR REPLACE**.

```
CREATE OR REPLACE VIEW dept_sal (COD, SAL_MEDIU)  
AS SELECT deptno, AVG(sal)  
FROM emp  
GROUP BY deptno;
```

The screenshot shows a SQL IDE interface. At the top, there's a 'SQL Commands' tab with a 'Schema' dropdown set to 'LUCRU'. Below this, there are controls for 'Language' (set to 'SQL'), 'Rows' (set to '20'), and buttons for 'Clear Command' and 'Find Tables'. The main area displays the SQL command being executed, with line numbers 1 through 6 on the left. The command is: `1 CREATE OR REPLACE VIEW dept_sal (COD, SAL_MEDIU)`, `2 AS SELECT deptno, AVG(sal)`, `3 FROM emp`, `4 GROUP BY deptno;`, `5`, `6`. Below the command area, there's a 'Results' tab selected, showing the message 'View created.' and the execution time '0.05 seconds'. Other tabs visible are 'Explain', 'Describe', 'Saved SQL', and 'History'.

```
1 CREATE OR REPLACE VIEW dept_sal (COD, SAL_MEDIU)
2 AS SELECT deptno, AVG(sal)
3 FROM emp
4 GROUP BY deptno;
5
6
```

Results Explain Describe Saved SQL History

View created.

0.05 seconds

Verificăm crearea vederii:

```
SELECT OBJECT_NAME, OBJECT_TYPE  
FROM USER_OBJECTS  
WHERE OBJECT_TYPE='VIEW';
```

```
1 SELECT OBJECT_NAME, OBJECT_TYPE  
2 FROM USER_OBJECTS  
3 WHERE OBJECT_TYPE='VIEW';  
4
```

OBJECT_NAME	OBJECT_TYPE
CUSTOMER_ORDER_PRODUCTS	VIEW
DEPT_SAL	VIEW
EBA_COUNTRIES_V	VIEW
EMP10	VIEW
EMP20	VIEW
EMP30	VIEW
EMP_DEPT_V	VIEW
PRODUCT_ORDERS	VIEW
PRODUCT_REVIEWS	VIEW
STORE_ORDERS	VIEW
STORE_ORDERS_STATUS	VIEW

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

Exercițiu

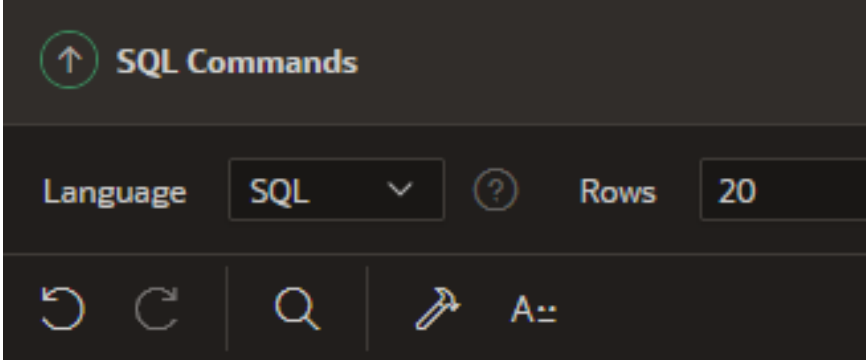
Vizualizați conținutul vederii cu o cerere **SELECT**

```
1 SELECT *
2 FROM dept_sal;
3
4
```

[illegible]

Modificați vederea **dept_sal** utilizând comanda **UPDATE**.

```
UPDATE dept_sal  
SET SAL_MEDIU = SAL_MEDIU + 1000  
WHERE cod = 10;
```



The screenshot shows a SQL editor window titled "SQL Commands". It has a "Language" dropdown set to "SQL", a "Rows" dropdown set to "20", and a toolbar with icons for undo, redo, search, and insert. Below the toolbar, the SQL command is entered in a monospaced font with syntax highlighting: line 1 is "UPDATE dept_sal", line 2 is "SET SAL_MEDIU = SAL_MEDIU + 1000", and line 3 is "WHERE cod = 10;". Line 4 is empty.

```
1  UPDATE dept_sal  
2  SET SAL_MEDIU = SAL_MEDIU + 1000  
3  WHERE cod = 10;  
4
```

Clauza WITH CHECK OPTION

Se folosește pentru a ne asigura că operația care se efectuează pe o vedere respectă domeniul vederii, adică nu va introduce modificări care vor determina ca liniile inserate să nu mai fie văzute de vedere.

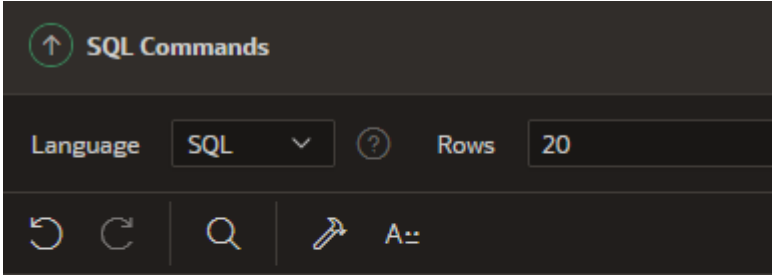
Exemplu

Să cream vederea **emp20** utilizând clauza **OR REPLACE**.

```
CREATE OR REPLACE VIEW emp20
AS SELECT *
FROM emp
WHERE deptno = 20
WITH CHECK OPTION CONSTRAINT dep20;
```

Apoi:

```
SELECT * FROM emp20;
```



The screenshot shows a SQL command window with a dark theme. At the top, it says 'SQL Commands'. Below that, there's a 'Language' dropdown set to 'SQL' and a 'Rows' field set to '20'. The command area shows the following SQL code:

```
1 CREATE OR REPLACE VIEW emp20
2 AS SELECT *
3 FROM emp
4 WHERE deptno = 20
5 WITH CHECK OPTION CONSTRAINT dep20;
```

The code is color-coded: 'CREATE OR REPLACE VIEW' is red, 'emp20' is blue, 'AS SELECT *' is blue, 'FROM emp' is blue, 'WHERE deptno = 20' is blue, and 'WITH CHECK OPTION CONSTRAINT dep20;' is red.

Încercăm să modificăm **sal_anual** într-un rând al vederii, adică să trecem cu ajutorul vederii o persoană din departamentul 20 în alt departament.

Ca să trecem efectiv o persoană dintr-un departament în altul ar trebui să ne asigurăm că aceea persoana există deci de aceea se vizualizează conținutul vederii **emp20**.

Am ales cod = 9999 dar ar putea fi oricare alt număr.

```
UPDATE emp20  
SET sal_anual = 70000  
WHERE cod = 9999
```

Notă:

Se observă că nu se poate schimba codul departamentului pentru că atunci vederea nu ar mai vedea respectiva persoană.

Deci cu ajutorul clauzei **WITH CHECK OPTION** vederea permite vizualizarea persoanelor din departamentul 20 și nu permite schimbarea codului pentru persoanele selectate în vedere.

Clauza WITH READ ONLY

*Cu ajutorul clauzei READ ONLY putem interzice operațiile **DML** printr-o vedere.*

Exemplu

Să cream o vedere **emp10** utilizând clauza **OR REPLACE**.

CREATE OR REPLACE VIEW emp10 (COD, NUME, FUNCTIE)
AS SELECT empno, ename, job
FROM emp
WHERE deptno = 10
WITH READ ONLY;

Apoi:

SELECT * FROM emp10;

SQL Commands

Schema: LUCRU

Language: SQL

Rows: 20

Clear Command Save Run

Find Tables

1 SELECT * FROM emp10;

2

3

4

5

Results Explain Describe Saved SQL History

COD	NUME	FUNCTIE
7839	KING	PRESIDENT
7782	CLARK	MANAGER
7934	MILLER	CLERK

3 rows returned in 0.01 seconds Download

SQL Commands

Language: SQL

Rows: 20

1 CREATE OR REPLACE VIEW emp10 (COD, NUME, FUNCTIE)

2 AS SELECT empno, ename, job

3 FROM emp

4 WHERE deptno = 10

5 WITH READ ONLY;

6

7

8

9

10

Results Explain Describe Saved SQL History

View created.

0.04 seconds

- Încercăm să modificăm vederea **emp10** alegând un angajat cu empno = 7839 dar acesta poate fi schimbat.

**DELETE FROM emp10
WHERE empno = 7839;**

- Orice încercare de a șterge sau modifica un rând al vederii se sfârșește prin eroare.

The screenshot shows an SQL IDE interface. At the top, the 'SQL Commands' tab is active, and the 'Schema' is set to 'LUCRU'. The 'Language' is 'SQL', and 'Rows' is set to 20. The command entered is:

```
1 DELETE FROM emp10
2 WHERE empno = 7839;
```

The 'Run' button is highlighted in green. Below the command editor, the 'Results' tab is active, displaying the following error messages:

```
ORA-00904: "EMPNO": invalid identifier
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 673
ORA-06512: at "SYS.DBMS_SYS_SQL", line 1658
ORA-06512: at "SYS.WWV_DBMS_SQL_APEX_210200", line 659
ORA-06512: at "APEX_210200.WWV_FLOW_DYNAMIC_EXEC", line 1829
```

Ștergerea unei vederi

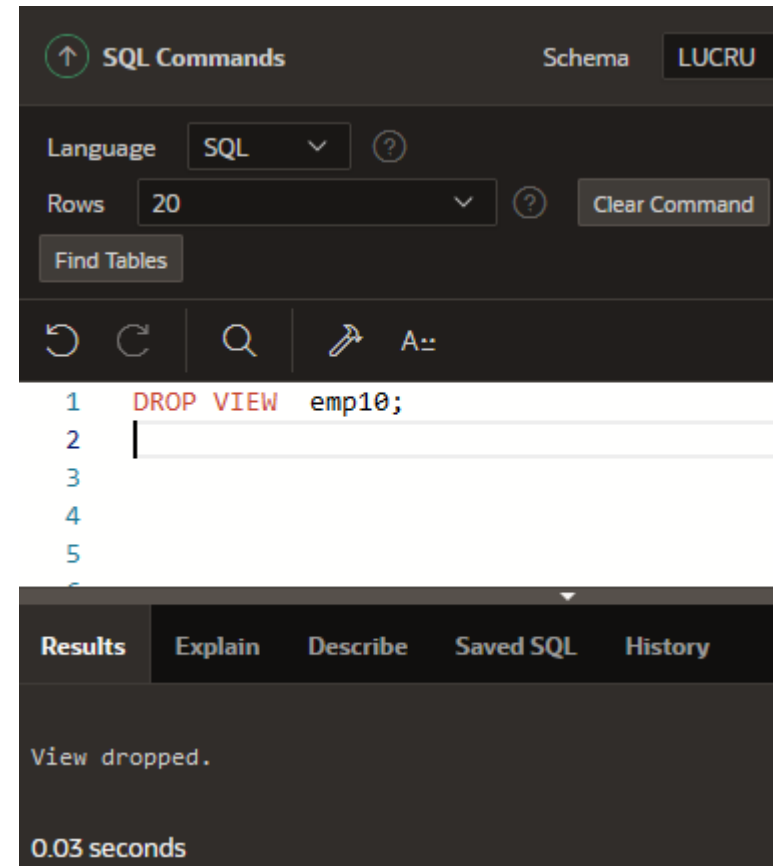
Sintaxa:

DROP VIEW view;

Exemplu

Distrugerea vederii **emp10**.

DROP VIEW emp10;



Verificăm că vederea nu mai există.

```
SELECT OBJECT_NAME, OBJECT_TYPE  
FROM USER_OBJECTS  
WHERE OBJECT_TYPE='VIEW';
```

```
1 SELECT OBJECT_NAME, OBJECT_TYPE
2 FROM USER_OBJECTS
3 WHERE OBJECT_TYPE='VIEW';
4
```

Results Explain Describe Saved SQL History				
OBJECT_NAME		OBJECT_TYPE		
CUSTOMER_ORDER_PRODUCTS		VIEW		
DEPT_SAL		VIEW		
EBA_COUNTRIES_V		VIEW		
EMP20		VIEW		
EMP30		VIEW		
EMP_DEPT_V		VIEW		
PRODUCT_ORDERS		VIEW		
PRODUCT_REVIEWS		VIEW		
STORE_ORDERS		VIEW		
STORE_ORDERS_STATUS		VIEW		

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

Cereri Top N

*Cererile **Top N** căuta cele mai mari n sau cele mai mici valori dintr-o coloană.*

Folosind cereri **Top N** putem afla de exemplu:

- primii 3 salariați dintr-o companie
- cele mai bine vândute 3 produse ale unei companii

Sintaxa

```
SELECT [ column_list ], ROWNUM  
FROM (SELECT [column_list]  
      FROM table  
      ORDER BY Top-N column)  
WHERE ROWNUM <= N;
```

Sintaxa conține:

O subcerere care întoarce elemente sortate și care conține clauza **ORDER BY**.

Ca să avem cea mai mare valoare pe prima linie folosim **DESC**.

O cerere care selectează un anumit număr de linii (N) ca rezultat final. Conține:

- **ROWNUM** care alocă o valoare începând de la 1 fiecărei linii întoarse de subcerere.
- **WHERE** care condiționează numărul de linii întoarse

Exemplu

Cele mari bine remunerate 2 persoane.

```
SELECT ROWNUM as pozitie, ename, sal  
FROM ( SELECT ename, sal  
        FROM emp  
        ORDER BY sal DESC )  
WHERE ROWNUM<=2;
```

```
1  SELECT ROWNUM as pozitie, ename, sal  
2      FROM ( SELECT ename, sal  
3              FROM emp  
4              ORDER BY sal DESC )  
5      WHERE ROWNUM<=2;  
6  
7  
8
```

Results	Explain	Describe	Saved SQL	History
POZITIE		ENAME		SAL
1		KING		5000
2		SCOTT		3000
2 rows returned in 0.00 seconds			Download	

Primele 3 persoane cu vechimea cea mai mare:

```
SELECT ROWNUM as pozitie_vechime, ename, hiredate  
FROM ( SELECT ename, hiredate  
        FROM emp  
        ORDER BY hiredate )  
WHERE ROWNUM <= 3;
```

```
1  SELECT ROWNUM as pozitie_vechime, ename, hiredate  
2      FROM ( SELECT ename, hiredate  
3              FROM emp  
4              ORDER BY hiredate )  
5      WHERE ROWNUM <= 3;  
6  
7  
8
```

Results

Explain

Describe

Saved SQL

History

POZITIE_VECHIME	ENAME	HIREDATE
1	SMITH	12/17/1980
2	ALLEN	02/20/1981
3	WARD	02/22/1981

3 rows returned in 0.00 seconds

[Download](#)

Întrebări?