

Sintaxa de bază a instrucțiunilor SQL

Obiective

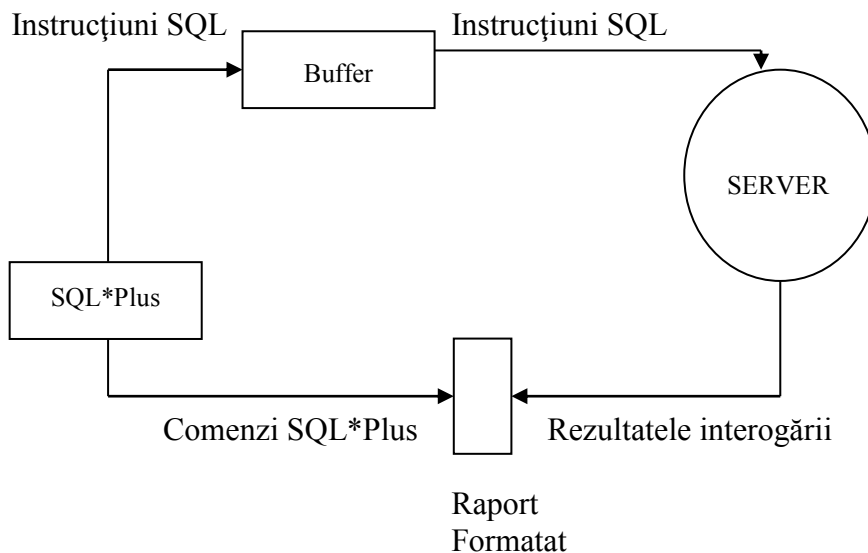
- Cunoașterea capacităților instrucțiunilor SELECT
- Executarea unor instrucțiuni SELECT de bază
- Cunoașterea diferențelor dintre instrucțiunile SQL și comenzile SQL*Plus

Introducere

SQL (Structured Query Language) este unul dintre limbajele relaționale de cereri care formează nucleul multor sisteme de gestiune a bazelor de date. El a fost dezvoltat pentru sistemul de gestiune al bazelor de date *System R* de cercetători din cadrul IBM Research Laboratory, San Jose-California la sfârșitul anilor '70. SQL a fost standardizat și a devenit limbajul de referință pentru bazele de date relaționale.

SQL este un limbaj de comenzi pentru comunicarea cu serverul Oracle din orice aplicație. Când se introduce o instrucțiune SQL, aceasta este stocată într-o zonă de memorie numită *buffer-ul SQL* și ea va rămâne în buffer până în momentul introducerii unei noi instrucțiuni.

SQL*Plus este un instrument Oracle care recunoaște și trimite instrucțiuni SQL la server pentru execuție și conține propriul limbaj de comenzi.



Caracteristicile limbajului SQL:

- poate fi folosit de o gamă largă de utilizatori, inclusiv de cei neprogramatori;
- este un limbaj neprocedural;
- reduce timpul necesar pentru crearea și întreținerea sistemelor.

Caracteristicile limbajului SQL*Plus:

- accepta fișiere de instrucțiuni SQL;
- furnizează un editor în mod linie pentru modificarea instrucțiunilor SQL;
- controlează setările de mediu;
- formatează rezultatele interogărilor sub formă de rapoarte;
- accesează baze de date locale și la distanță.

Următorul tabel cuprinde câteva elemente de comparație între SQL și SQL*Plus.

SQL	SQL*Plus
Este un limbaj pentru comunicarea cu serverul Oracle pentru accesarea datelor	Recunoaște instrucțiuni SQL și le trimite la server
Este bazat pe standardul ANSI SQL	Este o interfață proprietate Oracle pentru executarea instrucțiunilor SQL
Manipulează date și definiții de tabele în baze de date	Nu permite manipularea valorilor în baze de date
O instrucțiune se stochează în bufferul SQL pe una sau mai multe linii	Este permisă o comandă pe linie și nu este memorată în bufferul SQL
Nu are caracter de continuare	Folosește caracterul (-) pentru continuarea unei comenzi dacă aceasta nu încapă pe o linie
Folosește un caracter de terminare pentru executarea imediată a comenzii	Nu are nevoie de caractere de terminare. Se execută imediat
Nu pot fi abreviate	Pot fi abreviate
Folosește funcții pentru anumite formatare	Folosește comenzi pentru formatare

Mediul SQL*Plus oferă numeroase facilități, permițând:

- executarea instrucțiunilor SQL pentru extragerea, modificarea, adăugarea și ștergerea datelor din baza de date;
- formatarea, efectuarea de calcule, stocarea și listarea rezultatele interogărilor sub formă de rapoarte;
- crearea de fișiere de script pentru stocarea instrucțiunilor SQL.

Comenzile SQL*Plus se împart în următoarele mari categorii:

Categorie	Scop
Mediu	Afectează comportamentul general al instrucțiunilor SQL pentru sesiunea respectivă
Formatare	Formatează rezultatele interogării
Manipulări de fișiere	Salvează, încarcă și rulează fișiere de script
Execuție	Trimite instrucțiuni SQL din bufferul SQL la serverul Oracle
Editare	Modifică instrucțiunea SQL aflată în buffer
Interacțiune	Permite crearea și pasarea variabilelor către instrucțiuni SQL, afișarea variabilelor și listarea mesajelor pe ecran
Diverse	Diferite comenzi pentru conectarea la baza de date, manipularea mediului SQL*Plus și afișarea coloanelor

Pentru **conectarea într-un mediu de tip linie de comandă** se lansează următoarea comandă:

```
Sqlplus [username[/password[@database]]]
```

unde: username = numele utilizatorului din baza de date;

password = parola de conectare la baza de date;

@database = șirul de conectare la baza de date (numele bazei de date).

Notă: Din motive de securitate se recomandă introducerea numelui de utilizator și a numelui bazei de date, urmând ca apoi, la promptul `Password`, să se introducă parola.

Instrucțiunea SELECT

Pentru extragerea informațiilor din baza de date se utilizează instrucțiunea **SELECT**. Cu ajutorul acestei comenzi se pot realiza următoarele operații asupra datelor din baza de date:

- **SELECȚIE (SELECTION):** evaluarea rezultatului interogării va lua în calcul numai anumite linii din tabelele de date, pe baza unor criterii de selecție.
- **PROIECȚIE (PROJECTION):** rezultatul interogării va conține numai anumite coloane din tabele de date.
- **JONCȚIUNE (JOIN) :** utilizată pentru a combina date aflate în tabele diferite prin crearea unei legături între coloanele corespunzătoare.

Sintaxa de bază a comenzii **SELECT** este următoarea:

```
SELECT [ DISTINCT ] {*, coloana [alias] , ...}  
FROM tabel;
```

În cea mai simplă formă instrucțiunea **SELECT** conține:

- o clauză **SELECT** care specifică coloanele ce vor fi afișate;
- o clauză **FROM** care specifică tabelele ce conțin coloanele menționate în clauza **SELECT**.

Din punct de vedere sintactic:

SELECT este o listă de una sau mai multe coloane;

DISTINCT suprimă duplicatele;

***** selectează toate coloanele;

coloana numele coloanei;

alias redenumesc coloana selectată;

FROM tabel specifică tabelul ce conține coloanele din clauza **SELECT**.

Indicații privind scrierea instrucțiunilor SQL

Pentru a construi instrucțiuni valide, ușor de citit și de editat trebuie avute în vedere următoarele reguli:

- în general instrucțiunile SQL pot fi scrise cu litere mari sau mici;
- instrucțiunile SQL pot fi introduse pe una sau mai multe linii;

- cuvintele cheie nu pot fi abbreviate sau despărțite în linii diferite;
- de obicei clauzele sunt plasate pe linii separate pentru creșterea lizibilității textului;
- este de preferat ca toate cuvintele cheie să fie introduse cu majuscule, iar toate celelalte cuvinte, cum ar fi numele de tabele și coloane, să fie introduse cu litere mici;
- în mediul SQL*Plus, o instrucțiune SQL este introdusă la promptul SQL, iar următoarele linii aparținând instrucțiunii sunt numerotate. Acesta se numește buffer-ul SQL. Numai o singură instrucțiune SQL poate exista în buffer la un moment de timp.

Notă:

- un *cuvânt cheie* se referă la o clauză SQL, cum ar fi SELECT, FROM;
- o *clauză* reprezintă o parte a unei instrucțiuni SQL (e.g. SELECT empno, ename);
- o *instrucțiune* este dată de o combinație de una sau mai multe clauze (e.g. SELECT * FROM emp).

Execuția instrucțiunilor SQL se poate face prin:

- plasarea caracterului ';' la sfârșitul ultimei clauze;
- plasarea caracterului '/' la sfârșitul ultimei linii din buffer;
- plasarea unui caracter slash '/' la prompt-ul SQL;
- lansarea în SQL*Plus a comenzii **RUN** (la prompt-ul SQL).

Selectarea tuturor coloanelor și liniilor

Dacă se dorește afișarea tuturor coloanelor cu date dintr-un tabel se folosește cuvântul cheie SELECT împreună cu un asterix (*). În exemplul de mai jos, tabelul dept conține trei coloane (DEPTNO, DNAME și LOC) și patru linii, câte o linie pentru fiecare departament.

SELECT * FROM dept;

The screenshot shows a SQL*Plus window with the following elements:

- Top toolbar: Autocommit, Rows: 10, Save, Run.
- SQL prompt area: `SELECT * FROM dept;`
- Navigation tabs: Results (selected), Explain, Describe, Saved SQL, History.
- Table output:

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

4 rows returned in 0.04 seconds [Download](#)

Același rezultat se poate obține și dacă enumerăm toate coloanele după cuvântul cheie SELECT. De exemplu, următoarea instrucțiune SQL afișează toate coloanele și toate liniile din tabelul dept.

SELECT deptno, dname, loc FROM dept;

Autocommit Rows: 10 Save Run

```
SELECT deptno, dname, loc FROM dept;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

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

Selectarea unor anumite coloane

În cazul în care se dorește afișarea unor anumite coloane din tabel, se va specifica în clauza `SELECT` numele acelor coloane, separate prin virgulă (`,`). Ordinea în care coloanele selectate vor fi afișate pe ecran este dată de ordinea în care au fost enumerate în clauza `SELECT`. De exemplu, pentru a afișa toate departamentele și orașele din tabelul `dept` se va executa următoarea instrucțiune SQL:

```
SELECT deptno, loc FROM dept;
```

Autocommit Rows: 10 Save Run

```
SELECT deptno, loc FROM dept;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

DEPTNO	LOC
10	NEW YORK
20	DALLAS
30	CHICAGO
40	BOSTON

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

Afișarea rezultatului unei interogări respectă următoarele convenții:

- numele coloanelor de tip șir de caractere și dată calendaristică, precum și datele din tabel asociate coloanelor respective sunt aliniate la stânga;
- numele coloanelor de tip numeric și datele din coloanele respective sunt aliniate la dreapta;
- numele unei coloane de tip dată calendaristică sau șir de caractere poate fi trunchiat, dar capul de coloană de tip numeric nu poate fi trunchiat;
- numele coloanelor sunt afișate implicit cu litere mari; coloanele pot fi redenumite cu ajutorul unui alias.

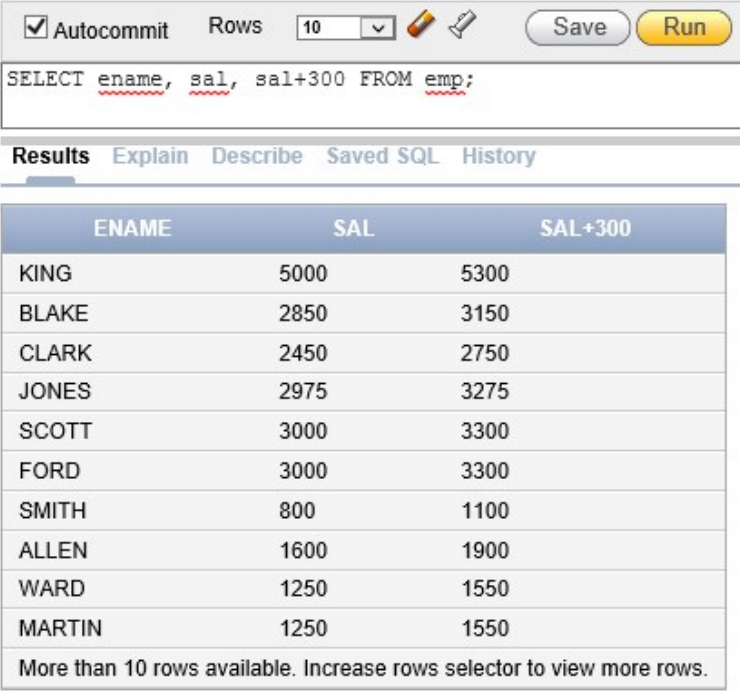
Expresii aritmetice

În cazul în care se dorește modificarea modului de afișare a datelor, în sensul executării unor calcule sau creării unor scenarii, se pot utiliza expresii aritmetice. Expresiile aritmetice pot conține

nume de coloane, constante numerice si operatori aritmetici (+, -, *, /). Acești operatori pot fi utilizați în orice clauză SQL, exceptând clauza FROM.

În exemplul următor se dorește afișarea numelui (ename), salariului (sal) și a valorii sal+300 pentru fiecare angajat din tabelul emp. Instrucțiunea SQL corespunzătoare este:

```
SELECT ename, sal, sal+300 FROM emp;
```



The screenshot shows a SQL query execution interface. At the top, there is a toolbar with a checked 'Autocommit' checkbox, a 'Rows' selector set to '10', and buttons for 'Save' and 'Run'. Below the toolbar, the SQL query is displayed: `SELECT ename, sal, sal+300 FROM emp;`. Underneath the query, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Results' tab is active, showing a table with three columns: 'ENAME', 'SAL', and 'SAL+300'. The table contains 10 rows of data. Below the table, there is a message: 'More than 10 rows available. Increase rows selector to view more rows.' and a 'Download' button. At the bottom, it says '10 rows returned in 0.01 seconds'.

ENAME	SAL	SAL+300
KING	5000	5300
BLAKE	2850	3150
CLARK	2450	2750
JONES	2975	3275
SCOTT	3000	3300
FORD	3000	3300
SMITH	800	1100
ALLEN	1600	1900
WARD	1250	1550
MARTIN	1250	1550

Exemplul anterior utilizează operatorul “+” pentru a mări salariile cu 300 pentru toți angajații și afișează o nouă coloană SAL+300. De notat că SAL+300 nu este o coloană nouă în tabelul emp. Implicit, denumirea noii coloane este dată de expresia aritmetică care a generat-o (în acest caz SAL+300).

Notă: SQL*Plus ignoră spațiile prezente înainte și după operatorii aritmetici.

Dacă o expresie aritmetică conține mai mult de un operator, înmulțirea și împărțirea sunt evaluate primele, apoi adunarea și scăderea. Dacă operatorii folosiți într-o expresie sunt de aceeași prioritate, evaluarea se va face de la stînga la dreapta. Folosirea parantezelor determină ca expresia dintre paranteze sa fie evaluată prima.

Valoarea null

Dacă o linie nu are date pentru o coloană particulară, această valoare se numește **null**. Semnificația valorii **null** este de dată nedisponibilă, neatribuită, necunoscută sau inaplicabilă. Valoarea **null** nu este aceeași cu valoarea 0 (pentru date de tip numeric) sau “” (pentru datele de tip șir de caractere).

Coloanele de orice tip pot conține valoarea null, cu excepția celor care au fost definite la creare ca fiind **NOT NULL** sau chei primare (**PRIMARY KEY**).

```
SELECT name, job, comm FROM emp;
```

Autocommit Rows: 30 Save Run

```
SELECT ename, sal, comm FROM emp;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

ENAME	SAL	COMM
KING	5000	-
BLAKE	2850	-
CLARK	2450	-
JONES	2975	-
SCOTT	3000	-
FORD	3000	-
SMITH	800	-
ALLEN	1600	300
WARD	1250	500
MARTIN	1250	1400
TURNER	1500	0
ADAMS	1100	-
JAMES	950	-
MILLER	1300	-

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

Se observă că în coloana COMM din tabelul emp angajatul TURNER poate avea comision, ceilalți angajați nefiind îndreptățiți să primească comision.

Dacă valoarea unei coloane ce apare într-o expresie aritmetică este null, rezultatul este null. După cum se știe, o încercare de împărțire la zero generează o eroare. În cazul unei împărțiri la o valoare null, rezultatul este null (necunoscut).

Definirea alias-urilor pentru coloane

La afișarea rezultatului unei interogări, SQL* Plus folosește numele coloanei ca și cap de tabel. Deoarece în multe cazuri acest cap de tabel nu este sugestiv, se poate schimba numele coloanei afișate folosind un *alias* de coloană. Alias-ul se specifică în clauza SELECT, imediat după numele coloanei. Dacă alias-ul conține spații, caractere speciale (\$) sau # - plasate la începutul alias-ului) sau dacă au importanță literele mari (mici) se va scrie alias-ul între ghilimele ("").

```
SELECT ename AS name, sal salary FROM emp;
```

Autocommit Rows: 10 Save Run

```
SELECT ename AS name, sal salary FROM emp;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

NAME	SALARY
KING	5000
BLAKE	2850
CLARK	2450
JONES	2975
SCOTT	3000
FORD	3000
SMITH	800
ALLEN	1600
WARD	1250
MARTIN	1250

More than 10 rows available. Increase rows selector to view more rows.

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

```
SELECT ename "Name", sal*12 "Annual Salary" FROM emp;
```

name	Salariul Anual
KING	60000
BLAKE	34200
CLARK	29400
JONES	35700
SCOTT	36000
FORD	36000
SMITH	9600
ALLEN	19200
WARD	15000
MARTIN	15000

More than 10 rows available. Increase rows selector to view more rows.

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

Primul exemplu afișează numele și salariul lunar al tuturor angajaților. Cuvântul cheie **opțional AS** a fost folosit înainte de alias, rezultatul fiind același dacă AS ar fi lipsit din clauza SELECT. De notat că alias-urile name și salary au fost scrise cu litere mici, iar la afișare au apărut scrise cu majuscule.

În cel de-al doilea exemplu se dorește afișarea numelor anagajaților și a salariilor anuale. Deoarece alias-ul Annual Salary implică folosirea spațiului, alias-ul trebuie scris între ghilimele (“ ”).

Operatorul de concatenare (||)

În cazul în care se dorește concatenarea unor coloane, a unor expresii aritmetice sau valori constante pentru a crea expresii de tip șir de caractere se va utiliza operatorul de concatenare (||). Coloanele ce apar de o parte și de alta a operatorului de concatenare sunt combinate pentru a forma o singură coloană la afișare.

```
SELECT ename || job AS "Employees" FROM emp;
```


Autocommit Rows: 10 Save Run

```
SELECT ename || job AS "Angajati" FROM emp;
```

Results Explain Describe Saved SQL History

Angajati
KINGPRESIDENT
BLAKEMANAGER
CLARKMANAGER
JONESMANAGER
SCOTTANALYST
FORDANALYST
SMITHCLERK
ALLENSALESMAN
WARDSALESMAN
MARTINSALESMAN
More than 10 rows available. Increase rows selector to view more rows.

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

În exemplul de mai sus, coloanele ENAME și JOB sunt concatenate, iar coloanei rezultante i se atribuie alias-ul `Employees`. Cuvântul cheie `AS` folosit înaintea alias-ului mărește lizibilitatea instrucțiunii `SELECT`.

Șiruri de caractere

Un “literal” este orice caracter, expresie sau număr inclus în lista `SELECT` care nu este un nume de coloană sau alias. Un literal este afișat pentru fiecare linie returnată de interogare. Valorile literale pentru datele calendaristice și caractere trebuie incluse între (‘ ’).

```
SELECT ename || ' ' || 'este' || ' ' || job AS "Employee Details" FROM emp;
```

Autocommit Rows: 10 Save Run

```
SELECT ename || ' ' || 'este' || ' ' || job AS "Detalii angajati" FROM emp;
```

Results Explain Describe Saved SQL History

Detalii angajati
KING este PRESIDENT
BLAKE este MANAGER
CLARK este MANAGER
JONES este MANAGER
SCOTT este ANALYST
FORD este ANALYST
SMITH este CLERK
ALLEN este SALESMAN
WARD este SALESMAN
MARTIN este SALESMAN
More than 10 rows available. Increase rows selector to view more rows.

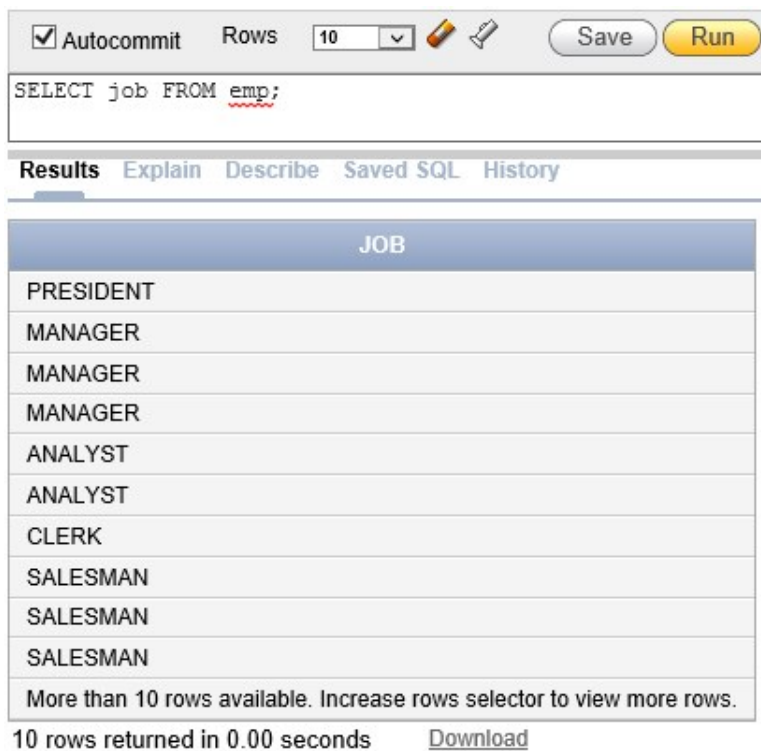
10 rows returned in 0.00 seconds [Download](#)

Exemplul de mai sus afișează numele și ocupația tuturor angajaților, numele coloanei fiind `Employee Details`. Spațiile dintre ghilimelele simple din instrucțiunea `SELECT` au fost introduse pentru îmbunătățirea lizibilității ieșirii.

Eliminarea rândurilor duplicat

În mod predefinit interogările afișează toate liniile, inclusiv cele duplicat.

```
SELECT job FROM emp;
```



Autocommit Rows 10 Save Run

```
SELECT job FROM emp;
```

Results Explain Describe Saved SQL History

JOB
PRESIDENT
MANAGER
MANAGER
MANAGER
ANALYST
ANALYST
CLERK
SALESMAN
SALESMAN
SALESMAN

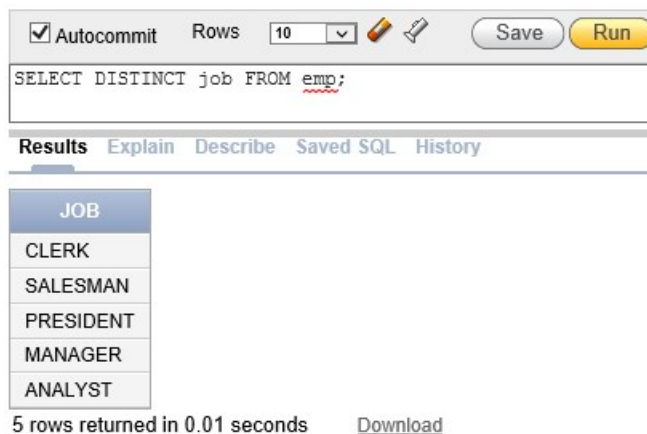
More than 10 rows available. Increase rows selector to view more rows.

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

SQL*Plus va afișa rezultatul interogării fără a elimina rândurile duplicat. Exemplul de mai sus afișează toate departamentele din tabelul `scott.emp`.

Eliminarea duplicatelor se face folosind cuvântul cheie **DISTINCT** în clauza `SELECT`.

```
SELECT DISTINCT job FROM emp;
```



Autocommit Rows 10 Save Run

```
SELECT DISTINCT job FROM emp;
```

Results Explain Describe Saved SQL History

JOB
CLERK
SALESMAN
PRESIDENT
MANAGER
ANALYST

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

În cazul în care se specifică mai multe coloane după clauza `DISTINCT` vor fi afectate toate coloanele selectate, rezultatul reprezentând o combinație distinctă a acestor coloane.

```
SELECT DISTINCT deptno, job FROM emp;
```

Comenzi PL*SQL

Afișarea structurii unui tabel

Se folosește comanda SQL*Plus `DESC[RIBE] tablename`, unde `tablename` este numele unui tabel, al unei vederi sau este un sinonim accesibil utilizatorului.

```
SQL> DESCRIBE dept
```

The screenshot shows the SQL*Plus command prompt with the command `DESCRIBE dept;` entered. Below the command, there are tabs for 'Results', 'Explain', 'Describe', 'Saved SQL', and 'History'. The 'Describe' tab is active, displaying the following table structure:

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DEPT	DEPTNO	NUMBER	-	2	0	1	-	-	-
	DNAME	VARCHAR2	14	-	-	-	✓	-	-
	LOC	VARCHAR2	13	-	-	-	✓	-	-

At the bottom right of the table structure, it indicates '1 - 3'.

Tabelul de mai sus conține informații privind structura tabelului `scott.dept`, numele coloanelor fiind enumerate în coloana **Name**. Coloana **Null?** indică dacă o coloană trebuie să conțină date. Valoarea `NOT NULL` semnifică faptul că valoarea `null` nu este permisă în acea coloană. `Type` reprezintă tipul de date al coloanei.

Tip de date	Descriere
NUMBER(p,s)	Valori numerice având un număr maxim de p cifre, unde s este numărul de cifre a părții subunitare
VARCHAR2(s)	Șir de caractere de lungime variabilă cu lungime maximă s
DATE	Date calendaristice între 1 ianuarie 4712 b.c. și 31 decembrie 9999 a.c.
CHAR(s)	Șir de caractere de lungime fixă s

Comenzile de editare ale mediului SQL*Plus

Câteva comenzi SQL*Plus utile în gestiunea buffer-ului SQL sunt:

- `A[PPEND]` `text` - adaugă `text` la sfârșitul liniei curente din buffer
- `C[HANGE]` `/ old / new` - schimbă textul `old` cu cel `new` în linia curentă
- `C[HANGE]` `/ text /` - șterge `text` din linia curentă din buffer
- `C[LEAR]` `BUFF[ER]` - șterge toate liniile din bufferul SQL
- `DEL` - șterge linia curentă din buffer
- `DEL n` - șterge linia `n` din buffer

- DEL m n - șterge din buffer toate liniile având numărul cuprins între m și n
- I[INPUT] - inserează un număr nedefinit de linii
- I[INPUT] text - inserează o linie în buffer, conținutul liniei fiind dat de text
- L[IST] - listează toate liniile din bufferul SQL
- L[IST] n - listează linia n din buffer
- L[IST] m n - listează liniile de la m la n din buffer
- n - linia n devine linia curentă din buffer
- n text - înlocuiește linia n din buffer cu text
- 0 text - inserează în buffer o linie înaintea liniei 1

Notă: La prompt-ul SQL se poate introduce doar o singură comandă SQL*Plus, ce nu este stocată în buffer. Pentru a continua o comandă pe linia următoare se va plasa caracterul (-) la sfârșitul liniei.

Comenzi SQL*Plus pentru fișiere

- SAVE filename [REP[LACE]APP[END]] - salvează conținutul buffer-ului SQL într-un fișier. Opțiunea APPEND adaugă la sfârșitul fișierului, iar opțiunea REPLACE suprascrie conținutul fișierului;
- GET filename - conținutul fișierului filename este încărcat în buffer-ul SQL (extensia implicită pentru fișier este .sql)
- START filename - rulează fișierul script filename
- @ filename - are același efect ca în cazul utilizării comenzii START
- ED[IT] - lansează editorul și salvează conținutul buffer-ului în fișierul afiedt.buf
- ED[IT] filename - lansează editorul pentru editarea conținutului unui fișier salvat
- SPO[OL] [filename] OFF | OUT - stochează rezultatul unei interogări într-un fișier. OFF închide fișierul. OUT închide fișierul și îl trimite la imprimantă sistem
- EXIT - ieșire din SQL*Plus

Probleme

1. Inițiați o sesiune SQL*Plus.
2. Comenzile SQL*Plus accesează o bază de date: adevărat / fals.
3. Următoarea instrucțiune SELECT se va executa cu succes : adevărat / fals.

```
SELECT ename, job, sal SALARY
FROM scott.emp;
```

4. Următoarea instrucțiune SELECT se va executa cu succes : adevărat / fals.

```
SELECT *
FROM scott.salgrade;
```

5. În instrucțiunea următoare sunt câteva erori. Identificați-le.

```
SELECT empno, ename
       Salary x 12 ANNUAL SALARY
FROM scott.emp;
```

6. Afișați structura tabelului scott.dept și conținutul lui.

7. Afișați structura tabelului `scott.emp`. Creați o interogare care să afișeze numele, funcția, data angajării și numărul angajatului pentru fiecare angajat, numărul angajatului apărând pe prima poziție. Salvați instrucțiunea într-un fișier `ex7.sql`.
8. Rulați interogarea din fișierul `ex7.sql`.
9. Afișați funcțiile (coloana `job`) din tabelul `scott.emp`, eliminând duplicatele.
10. Încărcați fișierul `ex7.sql` în buffer-ul SQL. Modificați instrucțiunea SQL din buffer astfel încât să redenumiți numele coloanelor cu `Emp#`, `Employee`, `Job`, `Hire Date` și apoi rulați interogarea.
11. Creați o interogare care să afișeze, pe o singură coloană, numele fiecărui angajat concatenat cu funcția sa, separate de virgulă și spațiu. Denumiți coloana `Employee and Title`.
12. Afișați datele din tabelul `scott.emp`, concatenând toate coloanele. Separați coloanele cu o virgulă și denumiți coloana `THE_OUTPUT`.

Rezultatul interogării este de forma:

```
THE_OUTPUT
-----
7369, SMITH, CLERK, 7902, 17-DEC-80, 800, , 20
7499, ALLEN, SALESMAN, 7698, 20-FEB-81, 1600, 300, 30
7521, WARD, SALESMAN, 7698, 22-FEB-81, 1250, 500, 30
.....
```