



Universitatea
Politehnica
București



FACULTATEA DE
**AUTOMATICĂ ȘI
CALCULATOARE**
Universitatea POLITEHNICA din București



Catedra de
Calculatoare

Laborator 2

Introducere în SQL și SQL*Plus

Autori

Conf. Dr. Ing. Alexandru Boicea

Ș.L. Dr. Ing. Ciprian-Octavian Truică



Cuprins

- SQL*Plus
- SQL
- Crearea unei Tabele
- Inserarea datelor într-o tabelă
- Vizualizarea datelor
- Ștergerea datelor



SQL*Plus

- SQL*Plus este un interpretor în linie de comandă (CLI – command line interpreter) care este utilizat pentru scrierea și lansarea către motorul bazei de date Oracle a interogărilor SQL și programelor PL/SQL;
- În mediul Windows, SQL*Plus apare și sub forma de aplicație GUI;
- Pentru a se face o conexiune la baza de date folosindu-se varianta GUI, un utilizator trebuie să introducă username, password și host string ;
- La instalarea bazei de date se pot instala schemele default care vin cu distribuția, cum ar fi schemele (userii) scott și hr.



SQL*Plus (Optional)

- Pentru a utiliza SQL*Plus în consolă, un utilizator trebuie să seteze, atât pe Windows cât și pe Linux, variabilele de mediu ORACLE_BASE, ORACLE_HOME, ORACLE_SID și să adauge la variabila de mediu PATH calea ORACLE_HOME/bin, și doar în Linux să adauge la LD_LIBRARY_PATH;
- ORACLE_BASE este variabila de mediu care indică directorul de bază unde este instalat software-ul ORACLE;
- ORACLE_HOME este variabila de mediu care indică directorul unde este instalat software-ul Oracle;
- ORACLE_SID este o variabilă de mediu care identifică o bază de date la nivelul sistemului de operare.



SQL*Plus (Optional)

- Setarea variabilelor de mediu în cmd/terminal:
- Windows (cmd)
 - set ORACLE_BASE=C:\app\oracle
 - set Oracle_HOME=%ORACLE_BASE%\product\12.1.0\dbhome_1
 - set ORACLE_SID=orcl
 - set PATH=%PATH%;%ORACLE_HOME%\bin
- Linux (terminal)
 - export ORACLE_BASE=/u01/app/oracle
 - export Oracle_HOME=\$ORACLE_BASE/product/12.1.0/dbhome_1
 - export ORACLE_SID=orcl
 - export PATH=\$ORACLE_HOME/bin:\$PATH
 - export LD_LIBRARY_PATH=\$ORACLE_HOME/lib:\$LD_LIBRARY_PATH



SQL*Plus

- Pentru a vă conecta la SQL*Plus în consolă comanda care se folosește este: `sqlplus [username[/password[@bazadate]]]`



```
[ciprian@bd-server1 ~]$ sqlplus
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Sat Feb 23 08:40:39 2019  
Version 18.3.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Enter user-name: student
```

```
Enter password:
```

```
Last Successful login time: Fri Jan 18 2019 20:48:11 +02:00
```

```
Connected to:
```

```
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production  
Version 18.3.0.0.0
```

```
SQL>
```



SQL*Plus



```
[ciprian@bd-server1 ~]$ sqlplus student
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Sat Feb 23 08:43:26 2019  
Version 18.3.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Enter password:
```

```
Last Successful login time: Sat Feb 23 2019 08:40:44 +02:00
```

```
Connected to:
```

```
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production  
Version 18.3.0.0.0
```

```
SQL>
```



SQL*Plus



```
[ciprian@bd-server1 ~]$ sqlplus student/student
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Sat Feb 23 08:44:31 2019  
Version 18.3.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Last Successful login time: Sat Feb 23 2019 08:43:29 +02:00
```

```
Connected to:
```

```
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production  
Version 18.3.0.0.0
```

```
SQL>
```




SQL*Plus



```
[ciprian@bd-server1 ~]$ sqlplus student/student@orcl
```

```
SQL*Plus: Release 18.0.0.0.0 - Production on Sat Feb 23 08:45:24 2019  
Version 18.3.0.0.0
```

```
Copyright (c) 1982, 2018, Oracle. All rights reserved.
```

```
Last Successful login time: Sat Feb 23 2019 08:43:29 +02:00
```

```
Connected to:  
Oracle Database 18c Enterprise Edition Release 18.0.0.0.0 - Production  
Version 18.3.0.0.0
```

```
SQL>
```



SQL*Plus

- Comenzile SQL*Plus pot fi editate direct în linia de comandă, sau pot fi editate cu orice editor cunoscut;
- Există un editor simplu, inclus în utilitar, care se accesează tastând **edit (ed)** în linia de comandă și care recunoaște următoarele comenzi:

Comandă	Abreviere	Utilizare
APPEND text	A text	Adaugă text la sfârșitul liniei curente
CHANGE/old/new	C/old/new	Schimbă old text cu new text în linia de comandă
CHANGE/text/	C/text/	Șterge textul din linia curentă
CLEAR BUFFER	CL BUFF	Șterge toate liniile din bufferul SQL



SQL*Plus

Comandă	Abreviere	Utilizare
DELETE	DEL	Șterge linia curentă
INPUT	I	Inserează un număr nedefinit de linii
INPUT text	I text	Inserează o linie continuă în textul respectiv
LIST	L	Listează toate liniile din buffer
LIST n	L n	Listează linia specificată
LIST n m	L n m	Listează liniile de la n la m
RUN	R	Afișează și execută comanda SQL curentă în buffer
	/	Execută comanda curentă din buffer



SQL*Plus

- Comanda editată rămâne în buffer până la editarea următoarei comenzi;
- O comandă poate fi editată pe mai multe linii tastând **ENTER** la sfârșitul liniei curente;
- Terminatorul pentru o comandă este **;**
- O comandă editată cu un editor se poate executa astfel:
 - Se copiază comanda din editor (Copy/Paste) și se execută cu **/** sau **RUN**;
 - Se execută tot scriptul (care poate conține mai multe comenzi) astfel:

```
SQL> @file_path_and_name
```

```
SQL> start file_path_and_name
```



SQL*Plus

- În linia de comandă se mai pot da și următoarele comenzi:
 - SAVE file_path_and_name – permite salvarea într-un fișier a conținutului bufferului SQL*Plus;
 - GET file_path_and_name – încarcă conținutul unui fișier salvat în prealabil în buffer;
 - START file_path_and_name – lansează în execuție comenzile salvate în fișier;
 - @file_path_and_name – execută comenzile din fișier;
 - EXIT – ieșire din SQL*Plus.



SQL

- Un sistem de gestionare a bazelor de date necesită un limbaj de interogare pentru a permite utilizatorilor să acceseze datele;
- SQL este un limbaj de interogare structurat care este utilizat de majoritatea bazelor de date relaționale;
- Caracteristici SQL:
 - Utilizează setul standard de comenzi: select, insert, update, delete, etc
 - Este un limbaj neprocedural, care optimizează cererile fără a specifica modul de execuție
 - Cererile se execută secvențial, înregistrare cu înregistrare, deci se prelucrează o singură înregistrare la un moment dat



SQL

- Comenzile SQL sunt conturate ca blocuri de interogare folosite pentru:
 - Inserarea, extragerea și ștergerea liniilor dintr-o tabelă
 - Crearea, modificarea și ștergerea obiectelor din baza de date
 - Controlul accesului la baza de date
 - Execuția calculelor
 - Specificarea alternativă a capetelor de coloane
 - Concatenarea coloanelor
 - Sortarea liniilor
 - Introducerea criteriilor de căutare



SQL

- Limbajul SQL are următoarele comenzi, grupate în 4 categorii:
 - Data Manipulation Language (DML): SELECT, INSERT, UPDATE, DELETE, MERGE
 - Data Definition Language (DDL): CREATE, ALTER, DROP, RENAME, TRUNCATE, COMMENT
 - Data Control Language (DCL): GRANT, REVOKE
 - Transaction Control Language (TCL): COMMIT, ROLLBACK, SAVEPOINT



SQL

- Comenzile DML
 - SELECT – folosită pentru a extrage datele din baza de date
 - INSERT – folosită pentru inserarea datelor în baza de date
 - UPDATE – folosită pentru modificarea datelor din baza de date
 - DELETE – folosită pentru ștergerea înregistrărilor
- Comenzile DDL
 - CREATE – folosită pentru crearea unui obiect (tabel, view, index, etc.) în baza de date
 - ALTER – folosită pentru modificarea structurii unui obiect din baza de date
 - DROP – folosită pentru ștergerea unui obiect din baza de date
- Comenzile DCL
 - GRANT – folosită pentru a grantifica drepturi de acces la un obiect din baza de date
 - REVOKE – folosită pentru a revoca drepturile de acces



SQL

- Reguli de scriere a comenzilor SQL
 - Comenzile se pot edita pe una sau mai multe linii;
 - Clauzele sunt uzual plasate pe linii separate;
 - Cuvintele predefinite nu pot fi separate pe mai multe linii;
 - Comenzile nu sunt *case sensitive*.



Crearea unei tabele

- Pentru a crea o tabelă se folosește comanda DDL CREATE TABLE.
- Sintaxa comenzii (versiunea simplificată):

CREATE TABLE [schema.] tablename

(
column_name_1 datatype [DEFAULT value | expression]
[inline_constraints] ,
[column_name_2 datatype [DEFAULT value | expression]
[inline_constraints] ,]
)



Crearea unei tabele

- **schema** – numele schemei în care se va crea tabelul
- **tablename** – numele tabelului
- **column_name_k** – numele coloanei k
- **datatype** – tipul de date al unei coloane
- **DEFAULT value | expression** – specific dacă o coloană va avea o valoare default, dată fie de o valoare fie de o expresie
- **inline_constraints** – o coloană poate să aibă constrângeri precum: UNIQUE, PRIMARY KEY, NOT NULL
- Mai multe detalii la adresa:

http://docs.oracle.com/cd/E16655_01/server.121/e17209/statements_7002.htm#SQLRF54469



Crearea unei tabele

- Ex. 1. Să se creeze un nomenclator de departamente, o tabelă pentru grila de salarizare și o tabelă pentru evidența angajaților.

```
1 CREATE TABLE departamente
2 (
3   id_departament NUMBER(2) NOT NULL,
4   denumire_departament VARCHAR2(30),
5   telefon VARCHAR2(10)
6 );
7
8 CREATE TABLE grila_salariu
9 (
10  grad NUMBER(2),
11  nivel_inferior NUMBER,
12  nivel_superior NUMBER
13 );
```



Crearea unei tabele



```
1 CREATE TABLE angajati
2 (
3   id_angajat NUMBER(4) NOT NULL,
4   nume VARCHAR2(30),
5   prenume VARCHAR2(30),
6   functie VARCHAR2(20),
7   id_sef NUMBER(4),
8   data_angajarii DATE DEFAULT SYSDATE,
9   salariu NUMBER(7,2) DEFAULT 2000,
10  comision NUMBER(7,2),
11  id_departament NUMBER
12 );
13
```



Crearea unei tabele

- Constrângerile sunt proprietăți ale unei coloane ce trebuie să fie îndeplinite în orice moment.
- Cele mai comune constrângeri sunt:
 - NOT NULL – valoarea unei coloane nu trebuie să fie NULL
 - UNIQUE – valoarea unei coloane (combinarea de valori ale unor coloane) trebuie să fie unice pentru fiecare înregistrare din tabelă
 - PRIMARY KEY – valoarea unei coloane (combinarea de valori ale unor coloane) trebuie să fie unice pentru fiecare linie din tabelă și să nu fie NULL. Cheia primară este identificatorul unic al unei înregistrări dintr-o tabelă
 - FOREIGN KEY – valoarea unei coloane (combinație de valori ale unor coloane) fac legătura cu alta tabela prin referirea cheii primare/unice din acea tabela



Crearea unei tabele

- Ex. 2. Să se adauge chei primare si chei străine pentru tabelele de la Ex. 1. Pentru tabela departamente să se adauge o constrângere unique pentru coloana denumire_departament. Pentru tabela angajati sa se adauge o constrângere unique pentru coloanele nume și prenume.



Crearea unei tabele



```
1 -- Primary Keys
2 ALTER TABLE departamente ADD CONSTRAINT pk_departamente PRIMARY KEY (id_departament);
3
4 ALTER TABLE grila_salariu ADD CONSTRAINT pk_grila PRIMARY KEY (grad);
5
6 ALTER TABLE angajati ADD CONSTRAINT pk_angajati PRIMARY KEY (id_angajat);
7
8 -- Foreign Keys
9 ALTER TABLE angajati ADD CONSTRAINT fk_ang__dep FOREIGN KEY (id_departament)
10 REFERENCES departamente(id_departament);
11
12 ALTER TABLE angajati ADD CONSTRAINT fk_ang__ang FOREIGN KEY (id_sef)
13 REFERENCES angajati(id_angajat);
14
15 -- Unique
16 ALTER TABLE departamente ADD CONSTRAINT uq_denumire UNIQUE (denumire_departament);
17
18 ALTER TABLE angajati ADD CONSTRAINT uq_nume UNIQUE (nume, prenume);
19
20 -- DROP TABLES
21 DROP TABLE angajati;
22 DROP TABLE departamente;
23 DROP TABLE grila_salariu;
```



Crearea unei tabele

- Obs. 1. Se pot crea toate constrângerile de la Ex. 2 în momentul când se creează tabelele (vezi următoarele slide-uri);
- Obs. 2. Atenție la ordinea în care se creează tabelele, deoarece nu se poate crea o cheie străină într-o tabelă atâta timp cât tabelă la care se face referință nu este creată;
- Obs. 3. În momentul când se dorește să se șteargă tabelele din baza de date, mai întâi trebuie șterse cele care referă alte tabele apoi tabelă care este referită prin cheie străină. Din exemplele noastre, mai întâi trebuie șterse tabelă **angajati** (care referă tabelă departamente printr-o cheie străină) și apoi tabelă **departamente**;
- Obs. 4. Pentru a vedea structura unei tabele se folosește comanda **desc[ribe] nume_tabel**, ex: **desc angajati**.



Crearea unei tabele



```
1 CREATE TABLE departamente
2 (
3   id_departament NUMBER(2) NOT NULL,
4   denumire_departament VARCHAR2(30),
5   telefon VARCHAR2(10),
6   CONSTRAINT pk_departamente PRIMARY KEY (id_departament),
7   CONSTRAINT uq_denumire UNIQUE (denumire_departament)
8 );
9
10 CREATE TABLE grila_salariu
11 (
12   grad NUMBER(2),
13   nivel_inferior NUMBER,
14   nivel_superior NUMBER,
15   CONSTRAINT pk_grila PRIMARY KEY (grad)
16 );
```



Crearea unei tabele



```
1 CREATE TABLE angajati
2 (
3   id_angajat NUMBER(4) NOT NULL,
4   nume VARCHAR2(30),
5   prenume VARCHAR2(30),
6   functie VARCHAR2(20),
7   id_sef NUMBER(4),
8   data_angajarii DATE DEFAULT SYSDATE,
9   salariu NUMBER(7,2) DEFAULT 2000,
10  comision NUMBER(7,2),
11  id_departament NUMBER,
12  CONSTRAINT pk_angajati PRIMARY KEY (id_angajat),
13  CONSTRAINT fk_ang__dep FOREIGN KEY (id_departament)
14    REFERENCES departamente(id_departament),
15  CONSTRAINT fk_ang__ang FOREIGN KEY (id_sef)
16    REFERENCES angajati(id_angajat),
17  CONSTRAINT uq_nume UNIQUE (nume, prenume)
18 );
```



Inserarea datelor într-o tabelă

- Pentru a insera informația într-o tabelă se va folosi comanda DML INSERT.
- Sintaxa comenzii (versiunea simplificată):
 - Specific în ce coloane vreau să inserez, trebuie să inserez în cel puțin coloanele care au constrângeri precum PRIMARY KEY și NOT NULL și nu au valori DEFAULT pe acestea

**INSERT INTO nume_tabel(column_name1 [, column_name2 [, ...]])
values(value1 [, value2, [...]])**

- Nu specific în ce coloane vreau să inserez, trebuie să dau valori pentru toate coloanele, în ordinea în care au fost declarate acestea

INSERT INTO nume_tabel values(value_column1, value_column2, ...)

- Mai multe detalii la adresa:

http://docs.oracle.com/cd/E16655_01/server.121/e17209/statements_9014.htm#i2111652



Inserarea datelor într-o tabelă

- Ex. 3. Să se insereze informații în tabelele create la Ex. 1. (Atenție la ordinea în care se face insertul)

```
1 INSERT INTO grila_salariu VALUES(1, 1, 1000);
2 INSERT INTO grila_salariu VALUES(1, 1001, 2000);
3
4 INSERT INTO departamente VALUES(10, 'Proiectare Software', '3212121');
5 INSERT INTO departamente VALUES(20, 'Asigurarea Calitatii', '3212122');
6 INSERT INTO departamente VALUES(10, 'Resurse Umane', '3212123');
7 INSERT INTO departamente VALUES(30, 'Proiectare Software', '3212124');
8
9 INSERT INTO angajati(id_angajat, nume, prenume, functie)
10 VALUES(1001, 'Ionescu', 'Ion', 'Director');
11 INSERT INTO angajati(nume, prenume, functie)
12 VALUES('Ionescu', 'Ionel', 'Programator');
13 INSERT INTO angajati(id_angajat, nume, prenume, functie, id_sef, salariu,
    id_departament)
14 VALUES(1002, 'Ionescu', 'Ionel', 'Programator', 1001, 1024, 10);
```



Vizualizarea datelor dintr-o tabelă

- Pentru a vizualiza datele dintr-o tabelă se folosește comanda DML SELECT.
- Sintaxa comenzii (versiunea simplificată)

```
SELECT [* | {column_name1[, column_name2[,...]]}]  
FROM table_name  
[WHERE conditions]
```

- Mai multe detalii la adresa

http://docs.oracle.com/cd/E16655_01/server.121/e17209/statements_10002.htm#SQLRF01702



Vizualizarea datelor dintr-o tabelă

- Ex. 4. Să se afișeze toate informațiile din tabela angajații.

```
1 SELECT * FROM angajati;
```

- Ex. 5. Să se afișeze numele, funcția și salariul pentru toți angajații.

```
1 column nume format a20  
2 column prenume format a20  
3 column functie format a20  
4 column salariu format 9999.99  
5 SELECT nume, prenume, functie, salariu  
6 FROM angajati;
```




Vizualizarea datelor dintr-o tabelă

- Ex. 6. Să se afișeze numele, prenumele și salariu pentru angajații din departamentul 10.

```
1 SELECT nume, prenume, functie, salariu  
2 FROM angajati  
3 WHERE id_departament = 10;
```

- Ex. 7. Să se afișeze numele, prenumele și salariu pentru angajații care au funcția 'Director'.

```
1 SELECT nume, prenume, functie, salariu  
2 FROM angajati  
3 WHERE functie='Director';
```



Modificarea datelor dintr-o tabelă

- Pentru a face o modificare se folosește comnda DML UPDATE
- Sintaxa comenzii (versiunea simplificată)

**UPDATE table_name SET column_name1 = new_value1 [,
column_name2 = new_value2, [...]]
[WHERE conditions]**

- Mai multe detalii la adresa

http://docs.oracle.com/cd/E16655_01/server.121/e17209/statements_10008.htm#SQLRF01708



Modificarea datelor dintr-o tabelă

- Ex. 8. Să se modifice numele angajatului cu id-ul 1002 în 'Popescu'. Verificați dacă s-a făcut modificarea.



```
1 SELECT nume FROM angajati WHERE id_angajat = 1002;  
2 UPDATE angajati SET nume = 'Popescu' WHERE id_angajat = 1002;  
3 SELECT nume FROM angajati WHERE id_angajat = 1002;
```

- Ex. 9. Să se acorde comision la toți angajații egal cu 10% din salariu. Verificați dacă s-a făcut modificarea.



```
1 SELECT nume, salariu, comision FROM angajati;  
2 UPDATE angajati SET comision = 0.1 * salariu;  
3 SELECT nume, salariu, comision FROM angajati;
```



Ștergerea înregistrărilor dintr-o tabelă

- Pentru a șterge o înregistrare se folosește comanda DML DELETE;

- Sintaxa comenzii (variantea simplificată)

DELETE FROM tablename [WHERE conditions];

- Mai multe detalii la adresa

http://docs.oracle.com/cd/E16655_01/server.121/e17209/statements_8005.htm#SQLRF01505



Ștergerea înregistrărilor dintr-o tabelă

- Ex. 10. Să se șteargă
 - Angajatul cu numele 'Pop' și prenumele 'Alina'
 - Angajații care a venit în firmă înainte de '1-JAN-1985'
 - Angajații care au comisionul null



Ștergerea înregistrărilor dintr-o tabelă

```
1 INSERT INTO angajati(id_angajat, nume,  
2   prenume, functie, id_sef,  
3   id_departament, data_angajarii)  
4 VALUES(4003, 'Pop', 'Alina', 'Analist',  
5   1001, 20, '21-NOV-1995');  
6  
7 INSERT INTO angajati(id_angajat, nume,  
8   prenume, functie, id_sef,  
9   id_departament, data_angajarii)  
10 VALUES(4004, 'Pop', 'Calin', 'Tester',  
11   1001, 20, '01-JAN-1980');  
12  
13 INSERT INTO angajati(id_angajat, nume,  
14   prenume, functie, id_sef,  
15   id_departament, data_angajarii)  
16 VALUES(4005, 'Pop', 'Andrei', 'Manager',  
17   1001, 20, '14-FEB-1995');
```



Ștergerea înregistrărilor dintr-o tabelă



```
1 SELECT nume, prenume, data_angajarii, comision
2 FROM angajati;
3
4 DELETE FROM angajati
5 WHERE nume = 'Pop' AND prenume = 'Alina';
6
7 SELECT nume, prenume, data_angajarii, comision
8 FROM angajati;
9
10 DELETE FROM angajati
11 WHERE data_angajarii < '01-JAN-1985';
12
13 SELECT nume, prenume, data_angajarii, comision
14 FROM angajati;
15
16 DELETE FROM angajati
17 WHERE comision IS NULL;
18
19 SELECT nume, prenume, data_angajarii, comision
20 FROM angajati;
```