

Laborator 1 – Interogarea datelor

1. Obiective

- Familiarizarea cu interfața DataGrip
- Cunoașterea capabilităților instrucțiunilor SELECT
- Executarea unor instrucțiuni SELECT de bază

Pentru detalii privind interfața DataGrip și modul de creare a unei scheme, vezi Anexa 1 (edu.tuiasi.ro).

2. 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.

2.1. Sintaxa de bază

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 din care se extrag coloanele din 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	redenumeste 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:

- instrucțiunile SQL pot fi scrise cu litere mari sau mici (case insensitive);
- 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;

Notă:

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

2.2. Selectarea coloanelor

2.2.1. 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 (*).

Exemplu: Tabelul `dept` de mai jos, conține trei coloane (`deptno`, `dname` și `loc`) și patru linii, câte o linie pentru fiecare departament.

```
SELECT *
FROM dept;
```

	deptno	dname	loc
1	10	ACCOUNTING	NEW YORK
2	20	RESEARCH	DALLAS
3	30	SALES	CHICAGO
4	40	OPERATIONS	BOSTON

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;
```

2.2.2. 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`.

Exemplu: pentru a afișa toate departamentele și orașele din tabelul `dept` se va executa următoarea instrucțiune SQL:

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

	deptno	loc
1	10	NEW YORK
2	20	DALLAS
3	30	CHICAGO
4	40	BOSTON

2.3. Eliminarea valorilor duplicate – DISTINCT

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

```
SELECT job
FROM emp;
```

	job
1	CLERK
2	SALESMAN
3	SALESMAN
4	MANAGER
5	SALESMAN
6	MANAGER
7	MANAGER
8	ANALYST
9	PRESIDENT
10	SALESMAN

Rezultatul interogării este afișat fără a elimina rândurile duplicat. Exemplul de mai sus afișează toate joburile din tabelul emp.

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

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

	job
1	CLERK
2	SALESMAN
3	MANAGER
4	ANALYST
5	PRESIDENT

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;
```

	deptno	job
1	20	CLERK
2	30	SALESMAN
3	20	MANAGER
4	30	MANAGER
5	10	MANAGER
6	20	ANALYST
7	10	PRESIDENT
8	30	CLERK
9	10	CLERK

2.4. Expresii aritmetice în SELECT

În cazul în care se dorește executarea unor calcule matematice, se pot utiliza expresii aritmetice. Expresiile aritmetice pot conține nume de coloane, constante numerice și operatori aritmetici (+, -, *, /). Acești operatori pot fi utilizați în orice clauză SQL, exceptând clauza FROM.

Exemplu: Să se afișeze numele (ename), salariul (sal) și valoarea sal+300 pentru fiecare angajat din tabelul emp. Instrucțiunea SQL corespunzătoare este:

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

	ename	sal	'sal+300'
1	SMITH	800	1100
2	ALLEN	1600	1900
3	WARD	1250	1550
4	JONES	2975	3275
5	MARTIN	1250	1550
6	BLAKE	2850	3150
7	CLARK	2450	2750
8	SCOTT	3000	3300
9	KING	5000	5300
10	TURNER	1500	1800

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`).

Dacă o expresie aritmetică conține mai mult de un operator, ordinea efectuării operațiilor este cea din algebră.

2.5. Alias-uri pentru coloane

La afișarea rezultatului unei interogări, SQL 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) acesta se va scrie între ghilimele (`" "`).

Exemplu: Să se afișeze 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`.

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

	name	salary
1	SMITH	800
2	ALLEN	1600
3	WARD	1250
4	JONES	2975
5	MARTIN	1250
6	BLAKE	2850
7	CLARK	2450
8	SCOTT	3000
9	KING	5000
10	TURNER	1500

Exemplu: se dorește afișarea numelor angajaților și a salariilor anuale. Deoarece alias-ul `Annual Salary` implică folosirea spațiului, alias-ul trebuie scris între ghilimele (`" "`).

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

	Name	Annual Salary
1	SMITH	9600
2	ALLEN	19200
3	WARD	15000
4	JONES	35700
5	MARTIN	15000
6	BLAKE	34200
7	CLARK	29400
8	SCOTT	36000
9	KING	60000

2.6. Concatenarea valorilor

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 funcția de concatenare (CONCAT).

CONCAT(expresie1, expresie2 [, expresie3,...])

Coloanele ce urmează una după alta în funcția de concatenare sunt combinate pentru a forma o singură coloană la afișare.

Exemplu: coloanele `ename` și `job` sunt concatenate, iar coloanei rezultante i se atribuie alias-ul `Employees`.

```
SELECT CONCAT(ename, job) AS "Employees"
FROM emp;
```

	Employees
1	SMITHCLERK
2	ALLENSALESMAN
3	WARDSALESMAN
4	JONESMANAGER
5	MARTINSALESMAN
6	BLAKEMANAGER
7	CLARKMANAGER
8	SCOTTANALYST
9	KINGPRESIDENT
10	TURNERSALESMAN

2.7. Literali în SELECT

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 apostrofuri(' ').

Exemplu: să se afișeze numele și ocupația tuturor angajaților, numele coloanei fiind `Employee Details`. Spațiile dintre apostrofuri din instrucțiunea `SELECT` au fost introduse pentru îmbunătățirea lizibilității ieșirii.

```
SELECT CONCAT(ename, ' is a ', job) AS "Employee Details"
FROM emp;
```

	'Employee Details'
1	SMITH is a CLERK
2	ALLEN is a SALESMAN
3	WARD is a SALESMAN
4	JONES is a MANAGER
5	MARTIN is a SALESMAN
6	BLAKE is a MANAGER
7	CLARK is a MANAGER
8	SCOTT is a ANALYST
9	KING is a PRESIDENT
10	TURNER is a SALESMAN

2.8. Valoarea NULL

Dacă o linie nu are date pentru o coloană particulară, această valoare se numește **null**. Semnificația valori **null** este de dată indisponibilă, neatribuită, necunoscută sau inaplicabilă. Valoarea **null** nu este aceeași cu valoarea 0 (pentru date de tip numeric) sau "" (șir vid - 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 ename, job, comm
FROM emp;
```

	ename	job	comm
1	SMITH	CLERK	<null>
2	ALLEN	SALESMAN	300
3	WARD	SALESMAN	500
4	JONES	MANAGER	<null>
5	MARTIN	SALESMAN	1400
6	BLAKE	MANAGER	<null>
7	CLARK	MANAGER	<null>
8	SCOTT	ANALYST	<null>
9	KING	PRESIDENT	<null>
10	TURNER	SALESMAN	0

Se observă că în coloana `comm` din tabelul `emp` doar angajații cu job-ul “SALESMAN” (agent de vânzări) pot 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).

2.9. Afișarea structurii unui tabel

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

```
DESCRIBE emp;
```

	Field	Type	Null	Key	Default	Extra
1	empno	int	NO	PRI	<null>	
2	ename	varchar(20)	YES		<null>	
3	job	varchar(20)	YES		<null>	
4	mgr	int	YES		<null>	
5	hiredate	date	YES		<null>	
6	sal	int	YES		<null>	
7	comm	int	YES		<null>	
8	deptno	int	YES	MUL	<null>	

3. Probleme propuse

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

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

2. În instrucțiunea următoare sunt câteva erori. Identificați-le și apoi rulați instrucțiunea corectă.

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

3. Afișați structura tabelului dept și conținutul lui.
4. Afișați structura tabelului emp. Creați o interogare care să afișeze numele, funcția, data angajării și numărul angajatului pentru fiecare angajat, numărul angajatului apărând pe prima poziție.
5. Modificați instrucțiunea SQL de la exercițiul anterior astfel încât să redenumiți numele coloanelor cu Emp#, Employee, Job, Hire Date și apoi rulați interogarea.
6. Afișați funcțiile (coloana job) din tabelul emp, eliminând duplicatele.
7. 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.
8. Afișați datele din tabelul emp, concatenând toate coloanele, mai puțin mgr și comm. Separați coloanele cu o virgulă și denumiți coloana THE OUTPUT.

Rezultatul interogării este de forma:

```
THE OUTPUT
-----
7369, SMITH, CLERK, 12/17/1980, 800, 20
7499, ALLEN, SALESMAN, 02/20/1981, 1600, 30
7521, WARD, SALESMAN, 02/22/1981, 1250, 30
.....
```