

Baze de date-Anul 2

Laborator 1 SQL

I. Introducere. Comanda SELECT. Cereri mono-relație

1. Ce este un sistem de gestiune a bazelor de date? Dați exemple.

Un sistem de gestiune a bazei de date (SGBD) este un produs software care asigură interacțiunea cu o bază de date, permițând definirea, consultarea și actualizarea datelor din baza de date.

2. Ce este SQL?

SQL (Structured Query Language) este un limbaj neprocedural pentru interogarea și prelucrarea informațiilor din baza de date.

Compilerul limbajului SQL generează automat o procedură care accesează baza de date și execută comanda dorită.

SQL permite atât definirea, prelucrarea și interogarea datelor, cât și controlul accesului la acestea. Comenzile SQL pot fi integrate în programe scrise în alte limbaje, de exemplu Cobol, C, C++, Java etc.

3. Care sunt limbajele SQL?

În funcție de tipul acțiunii pe care o realizează, instrucțiunile SQL se împart în mai multe categorii. Datorită importanței pe care o au comenzile componente, unele dintre aceste categorii sunt evidențiate ca limbaje în cadrul SQL, și anume:

- limbajul de definire a datelor (LDD) – comenzile CREATE, ALTER, DROP;*
- limbajul de prelucrare a datelor (LMD) – comenzile INSERT, UPDATE, DELETE, SELECT;*
- limbajul de control al datelor (LCD) – comenzile COMMIT, ROLLBACK.*

Pe lângă comenzile care alcătuiesc aceste limbaje, SQL cuprinde:

- instrucțiuni pentru controlul sesiunii;*
- instrucțiuni pentru controlul sistemului;*
- instrucțiuni SQL încapsulate.*

4. Analizați sintaxa simplificată a comenzii SELECT:

```
SELECT { [ {DISTINCT | UNIQUE} | ALL] lista_campuri | *}  
FROM [nume_schemă.]nume_obiect ]  
      [, [nume_schemă.]nume_obiect ...]  
[WHERE condiție_clauza_where]  
[START WITH condiție_clauza_start_with  
 CONNECT BY condiție_clauza_connect_by]  
[GROUP BY expresie [, expresie ...]  
  [HAVING condiție_clauza_having] ]  
[ORDER BY {expresie | poziție} [, {expresie | poziție} ...] ]  
[FOR UPDATE  
  [OF [ [nume_schemă.]nume_obiect.]nume_coloană  
    [, [ [nume_schemă.]nume_obiect.]nume_coloană] ...]  
  [NOWAIT | WAIT număr_întreg] ];
```

Un element din *lista_campuri* are forma: *expresie [AS] alias*.

Care dintre clauze sunt obligatorii?

5. Care sunt regulile de scriere a comenzilor SQL (acceptă abrevieri, e nevoie de caracter de terminare)? In instructiunea urmatoare sunt 3 erori. Care sunt acestea?

```
SQL> SELECT employee_id, last_name  
2      salary x 12 ANNUAL SALARY  
3      FROM employees;
```

Obs: ANNUAL SALARY este un alias pentru câmpul reprezentând salariul anual.

Dacă un alias conține *blank*-uri, el va fi scris obligatoriu între ghilimele. Altfel, ghilimelele pot fi omise. *Alias*-ul apare în rezultat, ca și cap de coloană pentru expresia respectivă. Doar cele specificate între ghilimele sunt *case-sensitive*, celelalte fiind scrise implicit cu majuscule.

II. Exerciții

1. a) Consultați diagrama exemplu HR (Human Resources) pentru lucrul în cadrul laboratoarelor SQL.

b) Identificați cheile primare și cele externe ale tabelelor existente în schemă, precum și tipul relațiilor dintre aceste tabele.

2. Să se inițieze o sesiune SQL*Plus folosind *user ID*-ul și parola indicate.

3. Să se listeze **structura** tabelelor din schema HR (*EMPLOYEES*, *DEPARTMENTS*, *JOBS*, *JOB_HISTORY*, *LOCATIONS*, *COUNTRIES*, *REGIONS*), observând tipurile de date ale coloanelor.

Obs: Se va utiliza comanda *DESC[RIBE] nume_tabel*.

4. Să se listeze **conținutul** tabelelor din schema considerată, afișând valorile tuturor câmpurilor.

Obs: *SELECT * FROM nume_tabel;*

5. Să se afișeze codul angajatului, numele, codul job-ului, data angajării. Ce fel de operație este aceasta (selecție sau proiecție)?

6. Să se listeze, cu și fără duplicate, codurile job-urilor din tabelul *EMPLOYEES*.

```
SELECT job_id FROM employees;
```

```
SELECT DISTINCT job_id FROM employees;
```

7. Să se afișeze numele concatenat cu job_id-ul, separate prin virgula și spațiu, și etichetați coloana "Angajat și titlu".

Obs: Operatorul de concatenare este "||". Șirurile de caractere se specifică între apostrofuri (NU ghilimele, caz în care ar fi interpretate ca alias-uri).

```
SQL> SELECT last_name|| ' , ' || job_id "Angajat si titlu"  
      FROM employees;
```

8. Creați o cerere prin care să se afișeze toate datele din tabelul *EMPLOYEES*. Separați fiecare coloană printr-o virgulă. Etichetați coloana "Informații complete".

9. Să se listeze numele și salariul angajaților care câștigă mai mult de 2850 \$.

```
SELECT last_name, salary  
FROM employees  
WHERE salary > 2850;
```

10. Să se creeze o cerere pentru a afișa numele angajatului și numărul departamentului pentru angajatul nr. 104.

11. Să se afișeze numele și salariul pentru toți angajații al căror salariu nu se află în domeniul 1500-2850\$.

Obs: Pentru testarea apartenenței la un domeniu de valori se poate utiliza operatorul *[NOT] BETWEEN valoare1 AND valoare2*.

12. Să se afișeze numele, job-ul și data la care au început lucrul salariații angajați între 20 Februarie 1987 și 1 Mai 1989. Rezultatul va fi ordonat crescător după data de început.

```
SELECT __, __, __
FROM __
WHERE __ BETWEEN '20-FEB-1987' __ '1-MAY-1989'
ORDER BY __;
```

13. Să se afișeze numele salariaților și codul departamentelor pentru toti angajații din departamentele 10 și 30 în ordine alfabetică a numelor.

```
SELECT __, __
FROM __
__ department_id IN (10, 30)
__;
```

Obs: Apartenența la o mulțime finită de valori se poate testa prin intermediul operatorului IN, urmat de lista valorilor între paranteze și separate prin virgule:

expresie IN (valoare_1, valoare_2, ..., valoare_n)

14. Să se afișeze numele și salariile angajaților care câștigă mai mult de 3500 \$ și lucrează în departamentul 10 sau 30. Se vor eticheta coloanele drept *Angajat* și *Salariu lunar*.

15. Care este data curentă? Afișați diferite formate ale acesteia.

Obs:

➤ Funcția care returnează data curentă este SYSDATE. Pentru completarea sintaxei obligatorii a comenzii SELECT, se utilizează tabelul DUAL:

```
SELECT SYSDATE
FROM dual;
```

➤ Datele calendaristice pot fi formate cu ajutorul funcției TO_CHAR(data, format), unde formatul poate fi alcătuit dintr-o combinație a următoarelor elemente:

Element	Semnificație
D	Numărul zilei din săptămână (duminica=1; luni=2; ...)
DD	Numărul zilei din lună.
DDD	Numărul zilei din an.
DY	Numele zilei din săptămână, printr-o abreviere de 3 litere (MON, THU etc.)
DAY	Numele zilei din săptămână, scris în întregime.
MM	Numărul lunii din an.
MON	Numele lunii din an, printr-o abreviere de 3 litere (JAN, FEB etc.)
MONTH	Numele lunii din an, scris în întregime.
Y	Ultima cifră din an
YY, YYY, YYYY	Ultimele 2, 3, respectiv 4 cifre din an.
YEAR	Anul, scris în litere (ex: <i>two thousand four</i>).
HH12, HH24	Orele din zi, între 0-12, respectiv 0-24.
MI	Minutele din oră.
SS	Secundele din minut.
SSSSS	Secundele trecute de la miezul nopții.

16. Sa se afiseze numele și data angajării pentru fiecare salariat care a fost angajat in 1987. Se cer 2 soluții: una în care se lucrează cu formatul implicit al datei și alta prin care se formatează data.

Varianta1:

```
SELECT first_name, last_name, hire_date
FROM employees
WHERE hire_date LIKE ('%87%');
```

Varianta 2:

```
SELECT first_name, last_name, hire_date
FROM employees
WHERE TO_CHAR(hire_date, 'YYYY')='1987';
```

Sunt obligatorii ghilimelele de la șirul '1987'? Ce observați?

17. Să se afișeze numele și job-ul pentru toți angajații care nu au manager.

18. Sa se afiseze numele, salariul si comisionul pentru toti salariatii care castiga comisioane. Sa se sorteze datele in ordine descrescatoare a salariilor si comisioanelor.

19. Eliminați clauza WHERE din cererea anterioară. Unde sunt plasate valorile NULL în ordinea descrescătoare?

20. Să se listeze numele tuturor angajatilor care au a treia literă din nume 'A'.

Obs: Pentru compararea șirurilor de caractere, împreună cu operatorul LIKE se utilizează caracterele *wildcard*:

- % - reprezentând orice șir de caractere, inclusiv șirul vid;
- _ (*underscore*) – reprezentând un singur caracter și numai unul.

21. Să se listeze numele tuturor angajatilor care au 2 litere 'L' in nume și lucrează în departamentul 30 sau managerul lor este 101.

22. Să se afiseze numele, job-ul si salariul pentru toti salariatii al caror job conține șirul "clerk" sau "rep" si salariul nu este egal cu 1000, 2000 sau 3000 \$. (operatorul NOT IN)

23. Să se afiseze numele, salariul si comisionul pentru toti angajatii al caror salariu este mai mare decat comisionul ($salary * commission_pct$) marit de 5 ori.