Baze de Date
Laborator 1

Introducere

1. Ce este o bază de date ? Dar un sistem de gestiune a bazelor de date? Daţi exemple.

- ***Baza de date*** *este un ansamblu structurat de date coerente, fără redundanță inutilă, care pot fi accesate în mod concurent de către mai mulți utilizatori.*
- *Un* ***sistem de gestiune a bazelor de date*** *(SGBD) este un produs software care asigură interacţiunea cu o bază de date, permiţând definirea, consultarea şi actualizarea datelor din baza de date.*

2. Ce este SQL?

- ***SQL*** *(Structured Query Language) este un* ***limbaj*** *neprocedural pentru interogarea şi prelucrarea informaţiilor din baza de date.*
  - Compilatorul limbajului *SQL* generează automat o procedură care accesează baza de date şi execută comanda dorită.
  - *SQL* permite:
    - definirea datelor (LDD)
    - prelucrarea şi interogarea datelor (LMD)
    - controlul accesului la date (LCD).
  - Comenzile *SQL* pot fi integrate în programe scrise în alte limbaje, de exemplu *Cobol*, *C*, *C++*, *Java* etc.

3. Ce este SQL*Plus? Comenzile *SQL*Plus* accesează baza de date ?

- ***SQL*Plus*** este un ***utilitar*** *Oracle*, având comenzi proprii specifice, care recunoaşte instrucţiunile *SQL* şi le trimite server-ului *Oracle* pentru execuţie.
  - Dintre funcţionalităţile mediului *SQL*Plus*, se pot enumera:
    - editarea, executarea, salvarea şi regăsirea instrucţiunilor *SQL* şi a blocurilor *PL/SQL*;
    - calculul, stocarea şi afişarea rezultatelor furnizate de cereri;
    - listarea structurii tabelelor.
  - Tabelul următor evidenţiază diferenţele dintre instrucţiunile *SQL* şi cele *SQL*Plus*:

SQL	***SQL*Plus***
Este un **limbaj** de comunicare cu *server*-ul *Oracle* pentru accesarea datelor.	Recunoaşte instrucţiunile *SQL* şi le transferă *server*-ului *Oracle*.
Se bazează pe **standardul *ANSI*** pentru *SQL*.	Este o interfaţă specifică sistemului *Oracle* pentru execuţia instrucţiunilor *SQL*.
Prelucrează date şi defineşte obiecte din baza de date.	Nu permite prelucrarea informaţiilor din baza de date.
Utilizează funcţii pentru a efectua formatări.	Utilizează comenzi pentru a efectua formatări.
Instrucţiunile nu pot fi abreviate.	Comenzile pot fi abreviate.
Nu are un caracter de continuare a instructiunilor scrise pe mai multe linii.	Acceptă „-“ drept caracter de continuare pentru comenzile scrise pe mai multe linii.
Caracterul de terminare a unei comenzi este “;”	Nu necesită caracter de terminare a unei comenzi.

4. Care sunt limbajele SQL?

- În funcţie de tipul acţiunii pe care o realizează, instrucţiunile *SQL* se împart în mai multe categorii. Datorită importanţei pe care o au comenzile componente, unele dintre aceste categorii sunt evidenţiate ca limbaje în cadrul *SQL*, şi 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*, *SAVEPOINT*.
- Pe lângă instructiunile care alcătuiesc aceste limbaje, *SQL* cuprinde si alte tipuri de instructiuni:
  - instrucţiuni pentru controlul sesiunii;
  - instrucţiuni pentru controlul sistemului;
  - instrucţiuni *SQL* încapsulate.

5. Analizaţi sintaxa simplificată a comenzii SELECT:

SELECT { [{***DISTINCT*** | ***UNIQUE***} | ***ALL***] *lista_campuri* | *}
FROM [*nume_schemă*.]*nume_obiect*]
[, [*nume_schemă*.]*nume_obiect* …]
[***WHERE*** *condiţie_clauza_where*]
[***START WITH*** *condiţie_clauza_start_with*
CONNECT BY *condiţie_clauza_connect_by*]
[***GROUP BY*** *expresie* [*, expresie* …]
[***HAVING*** *condiţie_clauza_having*]]
[***ORDER BY*** {*expresie* | *poziţie*} [, {*expresie* | *poziţie*} …]]
[***FOR UPDATE***
[***OF*** [[*nume_schemă*.]*nume_obiect*.]*nume_coloană*
[, [[*nume_schemă*.]*nume_obiect*.]*nume_coloană*] …]
[***NOWAIT*** | ***WAIT*** *număr_întreg*]];

6. Care sunt regulile de scriere a comenzilor SQL (acceptă abrevieri, e nevoie de caracter de terminare)?

7. In instructiunea urmatoare sunt erori. Care sunt acestea?

SQL> SELECT employee_id, last_name

*salary * 12 ANNUAL SALARY*

FROM employees;

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

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

Exerciţii

1. a) Consultaţi diagrama exemplu *HR* (Human Resources) pentru lucrul în cadrul laboratoarelor de baze de date.
   b) Identificaţi cheile primare şi cele externe ale tabelelor existente în schemă, precum şi tipul relaţiilor dintre aceste tabele.

2. 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*.

3. Să se listeze **conţinutul** tabelelor din schema considerată, afişând valorile tuturor câmpurilor. (*SELECT * FROM nume_tabel;*)

4. Să se afişeze codul angajatului, numele, codul job-ului, data angajării. Salvati instructiunea SQL într-un fişier numit **Laborator1.sql**.

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

SELECT job_id FROM employees;

SELECT DISTINCT job_id FROM employees;

SELECT UNIQUE job_id FROM employees;

Obs. DISTINCT = UNIQUE

6. Să se afişeze numele concatenat cu prenumele si cu job_id-ul, separate prin virgula și spatiu. Etichetați coloana “Detalii Angajat”.

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

SELECT last_name || ' , ' || first_name …

FROM employees;

7. Sa se listeze numele si salariul angajaţilor care câştigă mai mult de 2850.

8. Să se creeze o cerere pentru a afişa numele angajatului şi numărul departamentului pentru angajatul având codul 104.

9. Să se modifice cererea de la problema 7 pentru a afişa numele şi salariul angajaţilor al căror salariu nu se află în intervalul [1400, 24000].

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

9.1. Sa se afiseze numele, prenumele si salariul angajatilor al caror salariu este in intervalul [3000,7000] => utilizand **between**

9.2. Modificarea cererii de la punctual 9.1 fara a utiliza de aceasta data **between.**

10. Să se afişeze numele, job-ul şi data la care au început lucrul salariaţii angajaţi între 20 Februarie 1987 şi 1 Mai 1989. Rezultatul va fi ordonat crescător după data de început.

SQL> SELECT __, __, __

FROM __

WHERE __

ORDER BY __;

11. Să se afişeze numele salariaţilor şi codul departamentelor pentru toti angajaţii din departamentele 10 şi 30 în ordine alfabetică a numelor.

SQL> SELECT __, __

FROM ___

____ department_id IN (10, 30)

_____ _____;

Obs: Apartenenţa la o mulţime finită de valori se poate testa prin intermediul operatorului *IN*, urmat de lista valorilor (specificate între paranteze şi separate prin virgule):

expresie IN (valoare_1, valoare_2, …, valoare_n)

12. Să se modifice cererea de la problema 11 pentru a lista numele şi salariile angajatilor care câştigă mai mult de 1500 şi lucrează în departamentul 10 sau 30. Se vor eticheta coloanele drept *Angajat* si *Salariu lunar*.

13. Care este data curentă? Afişaţi diferite formate ale acesteia.

Obs:

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

SQL> SELECT SYSDATE

FROM dual;

- Datele calendaristice pot fi formatate cu ajutorul funcţiei ***TO_CHAR*(*data*, *format*)**, unde formatul poate fi alcătuit dintr-o combinaţie a următoarelor elemente:

Element	***Semnificaţie***
D	Numărul zilei din săptămâna (duminica=1; luni=2; …sâmbătă=6)
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: *two thousand four*).
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 nopţii.

14. Să se afișeze numele şi data angajării pentru fiecare salariat care a fost angajat în 1987. Se cer 2 soluţii: una în care se lucrează cu formatul implicit al datei şi alta prin care se formatează data.

Varianta1:

……

WHERE hire_date LIKE ('%87%');

Varianta 2:

……

WHERE TO_CHAR(hire_date, 'YYYY')='1987';

Sunt obligatorii ghilimelele de la şirul '1987'? Ce observaţi?

15. Să se afişeze numele şi job-ul pentru toţi angajaţii care nu au manager.

16. Să se afișeze numele, salariul și comisionul pentru toti salariații care câștigă comision. Să se sorteze datele în ordine descrescătoare a salariilor și comisioanelor.

SQL> SELECT ___, _____, _____

FROM _____

WHERE _____

ORDER BY _____;

17. Eliminaţi clauza *WHERE* din cererea anterioară. Unde sunt plasate valorile *NULL* în ordinea descrescătoare?

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

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

- % - reprezentând orice şir de caractere, inclusiv şirul vid;
- _ (*underscore*) – reprezentând un singur caracter şi numai unul.

19. Să se listeze numele tuturor angajatilor care au cel putin 2 litere ‘L’ in nume şi lucrează în departamentul 30 sau managerul lor este 102.

20. Să se afiseze numele, job-ul si salariul pentru toti salariatii al caror job conţine şirul “CLERK” sau “REP” și salariul nu este egal cu 1000, 2000 sau 3000 $. (operatorul *NOT IN*).