# Creerea si managementul tabelelor + Constrangeri + Alte obiecte de baza de date

1. Creati o tabela numita TAB1\_<nume> cu urmatoarele coloane (si caracteristici ale acestora): TAB1\_ID (cheie primara; numeric), nume (obligatorie; caractere), prioritate (neobigatorie; numeric) si una TAB2\_<nume> cu urmatoarele coloane (si caracteristici ale acestora): TAB2\_ID (cheie primara; numeric), nume\_detaliu (obligatorie; caractere).
2. Apoi modificati TAB2\_<nume> si ii mai adaugati o coloana numita TAB1\_ID de tip numeric. Creati un FK intre cele 2 tabele pe coloanele TAB1\_ID. Apoi creati o cheie unica in tabela TAB2\_<nume> pe coloanele nume\_detaliu si TAB1\_ID.
3. Creati o secventa denumita TAB\_<nume>\_SQ.
4. In tabela TAB1\_<nume> introduceti cel putin 3 constructori de masini si ordinea prioritatii lor (pentru coloana TAB1\_ID veti folosi secventa creata anterior).

Ex: Dacia, Honda, Audi

1. In tabela TAB2\_<nume> introduceti cel putin cate 2 marci de masini asociate fiecarui constructor (pentru coloana TAB2\_ID veti folosi secventa creata anterior).

Ex: Logan si Sandero asociate la Dacia;

Accord si Civic asociate la Honda;

TT si A4 asicoate la Audi;

1. Afisati marca si constructorul in ordinea prioritatii.
2. Testati constrangerile de pe cele 2 tabele cu inserturi ce produc declansarea lor.

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

…………………………………………………………………………………………………

# Interogarea datelor:

1. Query-ul urmator se va executa corect?

*Select last\_name, job\_id, salary as Sal*

*From employees;*

DA / NU

1. Scrieti un query care sa afiseze toate datele din tabela JOBS.
2. Gasiti 4 erori in query-ul urmator si rescrieti-l pentru a fi corect:

*Select employee\_id, last\_name*

*sal x 12 Salariu Anual*

*From employees;*

Err1:………………………………………………………………………………

Err2:………………………………………………………………………………

Err3:………………………………………………………………………………

Err4:………………………………………………………………………………

1. Scrieti un query care sa afiseze toate id-urile de departamente, din tabela EMPLOYEES, id-urile sa nu se repete.
2. Afisati numele si job\_id-ul angajatilor despartite prin virgula intr-o coloana numita Angajatul si Titulatura ( datele continute vor fi de forma: Gicu,IT\_PROG ).

# Restrictionarea si sortarea datelor:

1. Afisati last\_name si salary pentru angajatii (EMPLOYEES) care au salariul mai mare de 14000.
2. Afisati last\_name si salary pentru angajatii (EMPLOYEES) care au salariul cuprins intre 2000 si 14000 si ordonati lista descrescator dupa salariu.
3. Afisati last\_name pentru angajatii (EMPLOYEES) din departamentul 50.
4. Afisati last\_name, salary si commission\_pct (comision), pentru angajatii ce au comision. Ordonati lista in functie de comision crescator.
5. Afisati last\_name, job\_id si salary pentru angajatii cu job\_id ST\_CLERK si SA\_REP si salariul este diferit de valorile 2500, 3500 si 7000. Ordonati lista dupa nume descendent.

# Functii single-row:

1. Afisati printr-un query data curenta. Numele coloanei sa fie Data Curenta.
2. Afisati last\_name, salary si un nou salariu cu 15% mai mare. Coloana noului salariu sa se numeasca NEWSAL si sa contina doar numere intregi.
3. Afisati first\_name (nume) cu majuscule, lungimea numelor angajatilor si salary \* commission\_pct pentru angajatii cu numele ce incep cu litera J, A sau M. Pentru salarii sau comisioane nule, valoarea acestora va fi 1 pt comision si 0 pt salariu.
4. Afisati last\_name si o coloana ce va contine numarul de luni lucrate pana la Craciun 2002. Coloana se va numi VECHIME. Ordonati lista dupa VECHIME descrescator.
5. Selectati last\_name si salary pentru angajatii ce au comision si formatati coloana salary astfel incat salariile sa fie afisate sub forma:

1000000000

\*100000000

\*\*10000000

\*\*\*1000000

\*\*\*\*100000

\*\*\*\*\*10000

\*\*\*\*\*\*1000

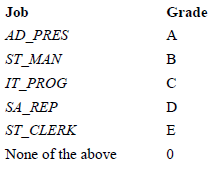
\*\*\*\*\*\*\*100

\*\*\*\*\*\*\*\*10

\*\*\*\*\*\*\*\*\*1

\*\*\*\*\*\*\*\*\*\*

, adica sa fie format din 10 caractere si caracterele lipsa (pana la lungimea de 10) de inaintea salariului sa fie inlocuite cu \*.

1. Scrieti un query prin care sa afisati job\_id si GARDE din tabela EMPLOYEES. GRADE va fi obtinut cu urmatoarea regula:

# Afisarea datelor din mai multe tabele:

1. Afisati printr-un query nume complet angajat (concatenare nume si prenume), numele departamentului in care lucreaza si prima zi din urmatoarea luna a zilei angajarii (HIRE\_DATE). Se vor afisa doar departamentele ce au manageri si se afla in Seattle.
2. Afisati printr-un query numele angajatului si numele sefului sau daca are unul, iar daca nu are in coloana NUME\_SEF pune-ti N/A. (folositi clauza JOIN in ambele standarde ANSI SQL 1999 si ORACLE).
3. Ce rezultat va intoarce queryul:

*Select \**

*From employees*

*cross join departments;*

……………………………………………………………………………………………..

# Agregarea datelor folosind funtii de grup:

1. Cati angajati se regasesc in tabla EMPLOYEES?
2. Determinati numarul de angajati pe departament.
3. Aflati salariul mediu, cel mai mic si cel mai mare salariu cat si suma tuturor salariilor pe departamente pentru angajatii ce sunt incadrati ca Sales Representative, Marketing Representative, Human Resources Representative sau Public Relations Representative. Lista va fi restrictionata doar la departamentele cu salariul mediu peste 7000.
4. Sintaxa sql se va executa corect:

*Select job\_id, count(1)*

*From jobs;*

DA / NU

De ce? ………………………………………………………………………………