

## **Baze de Date**

### **Informatică - Anul 1**

#### **Cerințe proiect – semestrul 2, 2020-2021**

Să se realizeze proiectarea, implementarea și exploatarea neprocedurală a unei baze de date relaționale, urmând pașii de mai jos:

1. Descrierea modelului real, a utilității acestuia și a regulilor de funcționare.
2. Prezentarea constrângerilor (restricții, reguli) impuse asupra modelului.
3. Descrierea entităților, incluzând precizarea cheii primare.
4. Descrierea relațiilor, incluzând precizarea cardinalității acestora.
5. Descrierea atributelor, incluzând tipul de date și eventualele constrângeri, valori implicite, valori posibile ale atributelor.
6. Realizarea diagramei entitate-relație corespunzătoare descrierii de la punctele 3-5.
7. Realizarea diagramei conceptuale corespunzătoare diagramei entitate-relație proiectate la punctul 6. Diagrama conceptuală obținută trebuie să conțină minimum 6 tabele (fără considerarea subentităților), dintre care cel puțin un tabel asociativ.
8. Enumerarea schemelor relaționale corespunzătoare diagramei conceptuale proiectate la punctul 7.
9. Realizarea normalizării până la forma normală 3 (FN1-FN3).
10. Crearea tabelelor în SQL și inserarea de date coerente în fiecare dintre acestea (minimum 5 înregistrări în fiecare tabel neasociativ; minimum 10 înregistrări în tabelele asociative).
11. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:
  - operație *join* pe cel puțin 4 tabele
  - filtrare la nivel de linii
  - subcereri sincronizate în care intervin cel puțin 3 tabele
  - subcereri nesincronizate în care intervin cel puțin 3 tabele
  - grupări de date, funcții grup, filtrare la nivel de grupuri
  - ordonări
  - utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a funcțiilor NVL și DECODE, a cel puțin unei expresii CASE
  - utilizarea a cel puțin 1 bloc de cerere (clauza WITH)
12. Implementarea a 3 operații de actualizare sau suprimare a datelor utilizând subcereri.
13. Crearea unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 10).

~~14. Crearea unei vizualizări compuse. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.~~

15. Crearea unui index care să optimizeze o cerere de tip căutare cu 2 criterii. Specificați cererea.
16. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația *outer-join* pe minimum 4 tabele și două cereri ce utilizează operația *division*.
17. Optimizarea unei cereri, aplicând regulile de optimizare ce derivă din proprietățile operatorilor algebrei relaționale. Cererea va fi exprimată prin expresie algebrică, arbore algebric și limbaj (SQL), atât anterior cât și ulterior optimizării.
18. a. Realizarea normalizării BCNF, FN4, FN5.  
b. Aplicarea denormalizării, justificând necesitatea acesteia.

## Observații

- Proiectul trebuie realizat **individual**.
- Proiectul trebuie realizat în sistemul **Oracle** (versiune minimă 11g) instalat pe calculatorul propriu.
- Un **proiect complex** presupune o diagramă conceptuală cu minimum 12 tabele (punctul 6), existența unei relații de tip superior (>2), tratarea complexă a fiecărei cerințe.
- Nu este necesară includerea tuturor atributelor în diagramele de la punctele 6 și 7; toate attributele vor fi enumerate obligatoriu în schemele relaționale (punctul 8).
- Diagrama conceptuală va fi proiectată în forma normală 3, astfel încât la punctul 9 să nu fie necesară o refacere a *design*-ului bazei de date, ci doar exemplificarea transformărilor de la fiecare formă normală, pe baza unor cazuri relative la model. Mai precis, se va da câte un exemplu de non-FN1, non-FN2, non-FN3 și transformarea fiecăruia în forma normală corespunzătoare. Este recomandabil ca rezultatul acestor normalizări să se regăsească în diagrama conceptuală (punctul 7).
- Pentru formele normale BCNF, FN4, FN5 se analizează modelul obținut la punctul 7 pentru a identifica relațiile care nu se află în aceste forme normale. Se va proceda într-unul dintre următoarele moduri:
  - Se aplică normalizarea în cazul identificat, fără a reproiecta modelul;
  - Se procedează asemănător observației anterioare (referitoare la FN1-FN3): se va da câte un exemplu de non-BCNF, non-FN4, non-FN5 și se vor aplica transformările pentru aducerea la BCNF, FN4, respectiv FN5. Exemplele vor fi relative la modelul proiectat anterior.

## Condiții de eligibilitate:

- Pentru a fi luat în considerare proiectul trebuie să conțină:

- un fișier doc/docx/pdf care să integreze toate rezolvările cerințelor, incluzând *print-screen*-uri prin care să se demonstreze că tot codul inclus în proiect a fost rulat în *Oracle*;
  - un fișier text care să conțină codul SQL de generare a bazei de date (comenzile de creare a tabelor și comenzile pentru inserarea datelor în aceste tabele);
  - un fișier text care să conțină codul SQL pentru cerințele de la punctele 11-14 (respectiv 11-16); se vor indica cerințele în limbaj natural sub formă de comentarii în cadrul acestui fișier, înaintea fiecărei implementări SQL.
- Cele 3 fișiere de mai sus vor fi denumite astfel: <Nume>\_<Prenume>-<tip\_document>.<extensie>, unde tip\_document va avea valorile "proiect", "creare\_inserare", "exemple" (de exemplu: Popescu\_Ana-proiect.docx, Popescu\_Ana-creare\_inserare.txt, Popescu\_Ana-exemple.txt). Fișierele astfel denumite vor fi încărcate până la termenul limită stabilit (**21 23 mai 2021, ora 23:59**), *link*-ul pentru încărcare fiind anunțat pe Moodle/Teams.