

Baze de Date

Informatică - Anul 1

Cerințe proiect – semestrul 2, 2024-2025

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 7 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 unei secvențe ce va fi utilizată în inserarea înregistrărilor în tabele (punctul 11).
 11. 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; maxim 30 de înregistrări în fiecare tabel).
 12. Formulați în limbaj natural și implementați 5 cereri SQL complexe ce vor utiliza, în ansamblul lor, următoarele elemente:
 - a) subcereri sincronizate în care intervin cel puțin 3 tabele
 - b) subcereri nesincronizate în clauza FROM
 - c) grupări de date, funcții grup, filtrare la nivel de grupuri cu subcereri nesincronizate (în clauza de HAVING)
 - d) ordonări și utilizarea funcțiilor NVL și DECODE (în cadrul aceleiași cereri)
 - e) utilizarea a cel puțin 2 funcții pe șiruri de caractere, 2 funcții pe date calendaristice, a cel puțin unei expresii CASE
 - f) utilizarea a cel puțin 1 bloc de cerere (clauza WITH)
- Observație:** Într-o cerere se vor regăsi mai multe elemente dintre cele enumerate mai sus, astfel încât cele 5 cereri să le cuprindă pe toate.
13. Implementarea a 3 operații de actualizare și de suprimare a datelor utilizând subcereri.

14. Crearea unei vizualizări complexe. Dați un exemplu de operație LMD permisă pe vizualizarea respectivă și un exemplu de operație LMD nepermisă.
15. Formulați în limbaj natural și implementați în SQL: o cerere ce utilizează operația *outer-join* pe minimum 4 tabele, o cerere ce utilizează operația *division* și o cerere care implementează analiza top-*n*.

Observație: Cele 3 cereri sunt diferite de cererile de la exercițiul 12.

16. La alegere: a) 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.
- sau b) Prezentarea planului de execuție a unei cereri complexe, optimizare/compare plan alternativ folosind hint-uri și obiecte specifice optimizării cererilor (spre exemplu indexi).
17. a. Realizarea normalizării BCNF, FN4, FN5.
- b. Aplicarea denormalizării, justificând necesitatea acesteia.
18. Exemplificarea *isolation levels* prin exemple de tranzacții care se execută în paralel în condiții de concurență, evidențiind efectele diferitelor niveluri de izolare asupra concurenței și integrității datelor.
19. Justificarea necesității/utilității migrării la o *bază de date de tip NoSql*. Identificarea scenariilor în care utilizarea unei baze de date NoSQL este mai avantajoasă decât a unei baze de date relaționale.
- a) Prezentarea structurii baze de date de tip NoSql.
 - b) Prezentarea comenzilor pentru crearea bazei de date (spre exemplu a colecțiilor într-o bază de date de tip document)
 - c) Prezentarea comenzilor pentru inserarea, modificarea și ștergerea documentelor sau înregistrărilor într-o bază de date NoSQL.
 - d) Exemplificarea comenzilor pentru interogarea datelor, incluzând operațiuni de filtrare și sortare..
20. Cerință **rezervată (flexibilă)** pentru alte concepte studiate relevant pentru dezvoltarea aplicațiilor cu suport pentru baze de date.

În rezolvarea proiectului se va ține seama de secțiunile următoare:

1. Specificații și observații importante (sistem și rezolvări cerințe)

- Proiectul trebuie realizat **individual**.
- Proiectul trebuie realizat obligatoriu în sistemul **Oracle** instalat pe calculatorul propriu. Recomandăm varianta *Enterprise Edition* (EE) disponibilă în versiunile 19c sau 21c: https://www.oracle.com/database/technologies/oracle-database-software-downloads.html#db_ee
Dacă versiunea EE nu poate fi instalată din diverse motive (de exemplu, cerințe de sistem ce nu pot fi satisfăcute), *Oracle XE (Express Edition)* este o variantă acceptabilă. Acesta este disponibil pentru download în versiunile 18c (<https://www.oracle.com/database/technologies/xel8c-downloads.html>) sau 21c (<https://www.oracle.com/ro/database/technologies/xe-downloads.html>).
- Pentru standardul minimal (nota 5) diagrama conceptuală trebuie să conțină **minimum 7 tabele** (fără a număra subentitățile), dintre care **minimum un tabel asociativ**.
- 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 atributele vor fi enumerate obligatoriu în schemele relaționale (punctul 8).
- **Denumirile tabelelor și ale atributelor vor fi obligatoriu în limba română.**
- **Diagramele vor fi realizate obligatoriu folosind regulile și notațiile prezentate la curs.**
- 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.
- Este **interzisă** obținerea codului sursă inclus în proiect prin generare automată cu ajutorul unor utilitare dedicate sau a platformelor bazate pe LLM.
- Este **interzisă** inserarea a mai mult de numărul maxim admis de înregistrări (30) în fiecare tabel.

2. Cerințe de redactare

Proiectul trebuie să conțină **obligatoriu**:

- Un fișier docx care să integreze toate rezolvările cerințelor și care să îndeplinească următoarele condiții:
 - va fi **structurat** conform cerințelor proiectului, va avea pagină de titlu și cuprins generat;
 - va include **print-screen-uri** prin care să se demonstreze că tot codul inclus în proiect a fost rulat în *Oracle* (cod vizibil, compilat fără erori);
 - va conține **tot codul SQL și sub formă de text** (nu doar ca imagine!);
 - pentru fiecare exercițiu va fi inclus **enunțul în limbaj natural**, codul SQL atât sub formă de text cât și ca *print-screen* din SQL Developer cu codul sursă (complet sau parțial) și rezultatul execuției acestuia;
 - pentru problema 11 *print-screen*-urile vor arăta că există date în tabelele create;
 - pentru fiecare exercițiu de la problema 12 se vor specifica elementele utilizate din lista prezentată în cadrul cerinței (subpunctele a-f).
- Un fișier text care să conțină codul SQL pentru cerințele de la punctele 10-11 (comenzile de creare a secvenței, a tabelor și comenzile pentru inserarea datelor în aceste tabele);
- Un fișier text care să conțină codul SQL pentru cerințele de la punctele 12-13 (respectiv 12-15); se vor indica cerințele în limbaj natural și elementele SQL folosite sub formă de comentarii în cadrul acestui fișier, înaintea fiecărei implementări SQL.

Cele 3 fișiere de mai sus **vor fi denumite** conform următorului format: <grupa>_<Nume>_<Prenume>-<tip_document>.<extensie>, unde *tip_document* va avea valorile "proiect", "creare_inserare", "exemple" (de exemplu: 131_Popescu_Ana-proiect.pdf, 131_Popescu_Ana-creare_inserare.txt, 131_Popescu_Ana-exemple.txt). Fișierele astfel denumite vor fi încărcate până **la termenul limită stabilit**, la *link*-ul pentru încărcare fiind anunțat (a se consulta fișierul *date_importante.txt* de pe Teams).

- Subiectele 18, 19, 20 vor fi tratate (opțional) în fișiere separate.

3. Condiții de eligibilitate

Pentru a fi luat în considerare, proiectul trebuie să fie conform:

- standardelor din fișierul *Modalitate_notare.pdf* (disponibil pe Teams) și
- tuturor cerințelor marcate ca obligatorii în secțiunile anterioare. **Nerespectarea oricărei condiții obligatorii conduce la neeligibilitatea proiectului.**

Reiterăm condiția de respectare a Regulamentului de etică și profesionalism (<https://drive.google.com/file/d/1gw2Fy44KnqaBKqLGh70vZPo3T9-ZdrDi/view>). Ca urmare a acestuia, sunt interzise acțiuni precum: copierea de la colegi (inclusiv cei din anii anteriori) sau din alte surse, generarea de cod cu ajutorul utilităților sau platformelor bazate pe LLM etc.

Dacă identificați un set de date care prezintă interes, o aplicație a cărei funcționalitate poate fi interesantă din perspectiva modelării bazei de date din backend (fără a o cunoaște pe cea reală) puteți indica acest lucru ca referință bibliografică. Aceasta permite preluarea și prelucrarea unor idei din perspectivă proprie, dar nu copierea și însușirea lor!

