Nume Prenume: Hororoi Valentin Alexandru

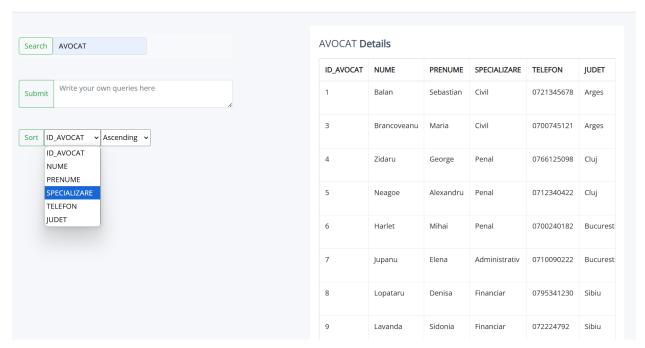
Grupa: 261

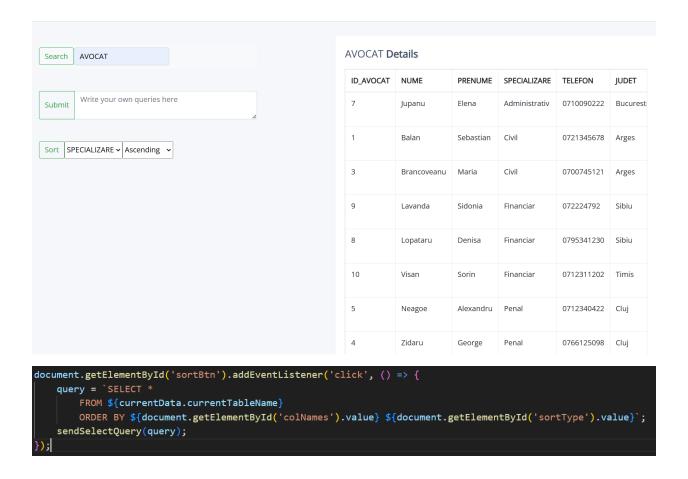
Tema proiect:Gestionarea Cazurilor Juridice

Rezolvări capitolul III

a) Listare conținut cu posibilitatea de sortare (toate tabelele).

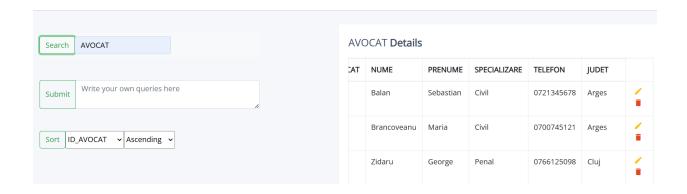
Sortarea are loc prin intermediul meniului de tip "dropdown", unde putem selecta numele coloanei și tipul sortării. În urma apăsării butonului "Sort", se trimite un query către baza de date, iar tabelul obținut în urma query-ului este afișat în interfață.





b) Modificare informații (opțiune de edit/ștergere pentru fiecare înregistrare).

Ultima coloană a tabelului conține 2 butoane, unul pentru editat și celălalt pentru șters. Ultimul screenshot surprinde toate query-urile trimise către baza de date în urma modificării unui atribut, iar mai apoi ștergerea întregii linii din acel tabel. Toate funcțiile sendUpdateQuery() sunt trimise cu parametrul "commit=true". Acest lucru se poate modifica pentru "commit=false", dar doar în cadrul codului (nu în interfață). Nu am mai implementat posibilitatea utilizatorului să aleagă daca modificările/ștergerile sunt cu/fără commit.

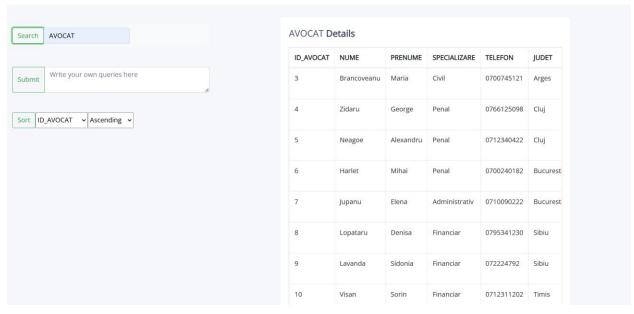


AVOCAT Details



AVOCAT Details

CAT	NUME	PRENUME	SPECIALIZARE	TELEFON	JUDET	
	EDITAT	Sebastian	Civil	0721345678	Arges	/ ■



```
Received update query: UPDATE AVOCAT

SET NUME = 'EDITAT', PRENUME = 'Sebastian', SPECIALIZARE = 'Civil', TELEFON = '0721345678', JUDET = 'Arges'
WHERE ID_AVOCAT = '1'

committed
Received select query: SELECT *
FROM AVOCAT

Received update query: DELETE FROM AVOCAT A
WHERE A.NUME = 'EDITAT' AND A.PRENUME = 'Sebastian' AND A.SPECIALIZARE = 'Civil' AND A.TELEFON = '0721345678' AND A.JUDET = 'Arges'

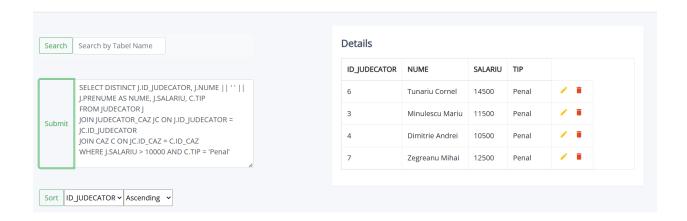
committed
Received select query: SELECT *
FROM AVOCAT
```

c) Afișarea rezultatului unei cereri care extrage informații din cel puțin 3 tabele și le filtrează cu ajutorul a cel puțin 2 condiții.

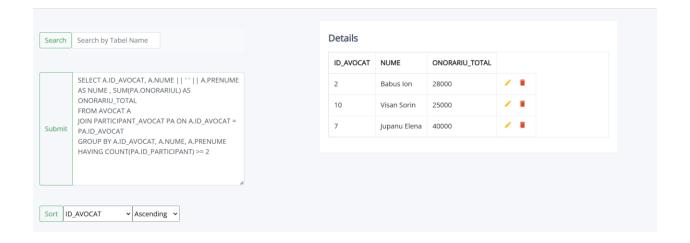
Am adăugat în interfață o zonă text, în urma apăsării butonului "Submit", textul este trimis ca query către baza de date. Daca query ul este scris corect și returnează informații, acestea sunt afișate în tabelul din interfață. Am trimis o cerere ce reunește tabelele JUDECATOR, JUDECATOR_CAZ, CAZ și returnează toți judecătorii care au salariul mai mare de 10000 și au lucrat la un caz de tip "Penal".

Query:

SELECT DISTINCT J.ID_JUDECATOR, J.NUME || ' ' || J.PRENUME AS NUME, J.SALARIU, C.TIP FROM JUDECATOR J
JOIN JUDECATOR_CAZ JC ON J.ID_JUDECATOR = JC.ID_JUDECATOR JOIN CAZ C ON JC.ID_CAZ = C.ID_CAZ
WHERE J.SALARIU > 10000 AND C.TIP = 'Penal'



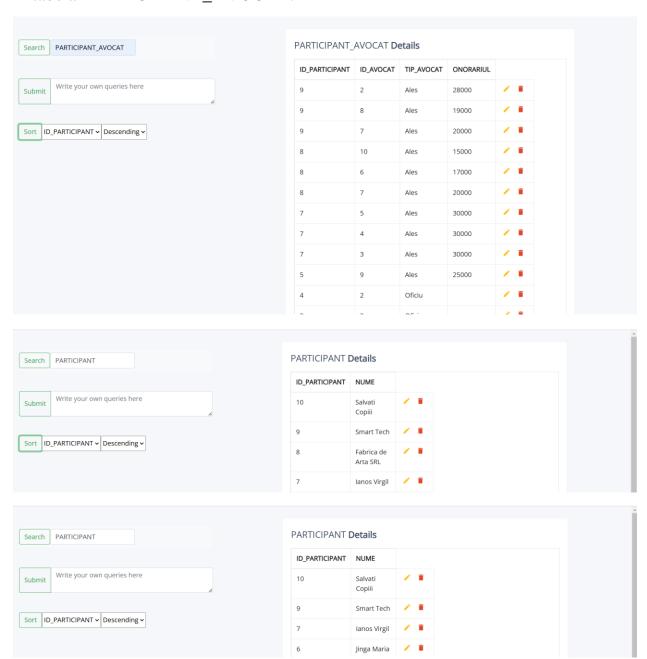
d) Afișarea rezultatului unei cereri care folosește funcții grup și o clauză *having* SELECT A.ID_AVOCAT, A.NUME || ' ' || A.PRENUME AS NUME , SUM(PA.ONORARIUL) AS ONORARIU_TOTAL FROM AVOCAT A JOIN PARTICIPANT_AVOCAT PA ON A.ID_AVOCAT = PA.ID_AVOCAT GROUP BY A.ID_AVOCAT, A.NUME, A.PRENUME HAVING COUNT(PA.ID_PARTICIPANT) >= 2

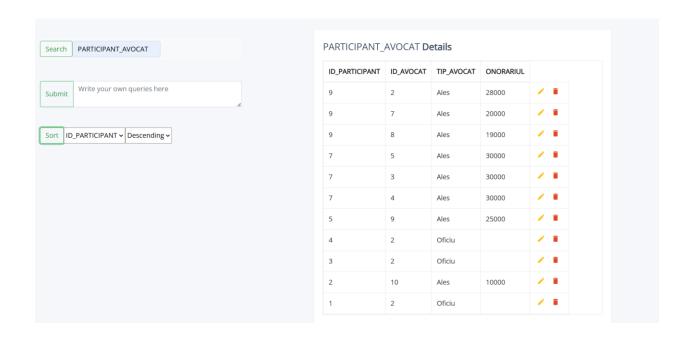


e) Implementarea unei constrângeri de tipul *on delete cascade* și exemplificare din interfață.

În urma ștergerii unei înregistrări al cărei cheie primară este cheie secundară în alt tabel, toate înregistrările care conțin referință către înregistrarea ștearsă, se vor șterge la randul lor. În exemplul de mai jos am șters înregistrarea cu

ID_PARTICIPANT=8, și am observat comportamentul ON DELETE CASCADE în tabela PARTICIPANT_AVOCAT.





f) Utilizarea vizualizărilor (cel puțin 2 vizualizări: *compusă* care să permită operații LMD, respectiv *complexă*)