Universitatea Tehnică a Moldovei

Facultatea - Calculatoare Informatică și Microelectronică

**Raport**

**la Baze de Date**

Lucrarea de laborator Nr.2

Tema: Crearea și modificarea tabelelor în SQL

A efectuat: st.gr. SI-211 Chirita Stanislav

A verificat: Rodica Bulai

Chișinău 2023

# 1 Sarcina lucrării

Aplicarea instrucțiunilor asupra tabelelor datelor de bază.

2 Obiective

Tratarea problemelor cu privire la crearea tabelelor, definirea coloanelor și setarea proprietăților de bază ale acestora, setarea proprietăților extinse ale coloanelor, stabilirea cheilor primare și celorlalte constrângeri de integritate, precum și modificarea structurii unui tabel.

3 Desfășurarea lucrării de laborator

1. Aflați toate datele despre grupele de studii de la facultate.
2. Sa se obtina lista disciplinelor in ordine descrescatoare a numarului de ore.
3. Aflați cursurile (Disciplina) predate de fiecare profesor (Nume\_Profesor, Prenume\_Profesor) sortate descrescător după nume și apoi prenume.
4. Afișați care dintre discipline au denumirea formată din mai mult de 20 de caractere?
5. Sa se afișeze lista studenților al căror nume se termina în ,,u" .
6. Afișati numele și prenumele primilor 5 studenti, care au obtinut note in ordine descrescatoare la al doilea test de la disciplina Baze de date. Sa se foloseasca optiunea TOP ... WITH TIES.
7. În ce grupa (Cod\_ Grupa) invata studentii care locuiesc pe strada 31 August?
8. Obtineti identificatorii și numele studentilor, ale examenelor sustinute in anul 2018.
9. Gasiti numele, adresa studentilor și codul disciplinei la care studentii au avut eel putin o nota mai mare decat 8 in 2018.
10. Gasiti studentii (numele, prenumele), care au obtinut la disciplina Baze de date (examen), in anul 2018, vreo nota mai mica de 8 și mai mare ca 4.
11. Furnizati numele și prenumele profesorilor, care au predat disciplina Baze de date, in 2018, și au evaluat vreun student cu nota nesatisfacatoare la reușita curenta.
12. Aflati cursurile urmate de catre studentul Florea loan.
13. Aflati numele și prenumele studentilor, precum și cursurile promovate cu note mai mari de 8 la examen.
14. Gasiti numele și prenumele studentilor, care au sustinut examen atat la profesorul Ion, cat și la profesorul Gheorge in anul 2017 (folositi pentru nume clauza LIKE)
15. Furnizați numele și prenumele studenților, care au studiat discipline cu un volum de lecții mai mic de 60 de ore, precum și profesorii (identificatorii) respectivi, care le-au predat.
16. Afișați numele și pronumele profesorilor, care au predat cel puțin o disciplina studentului cu identificatorul 100.
17. Afișați numele și prenumele profesorilor, care au predat doar discipline cu o încărcare orara mai mica de 60 de ore.
18. Găsiți numele și prenumele profesorilor, care au predat discipline, in care studentul "Cosovanu" a fost respins (nota < 5) la cel puțin o probă.
19. Afișați numărul de studenți care au susținut testul (Testul 2) la disciplina Baze de date in 2018.
20. cate note are fiecare student? Furnizați numele și prenumele lor.
21. Sa se obțină numărul de discipline predate de fiecare profesor (Nume\_Profesor, Prenume \_ Profesor).
22. Sa se obțină lista disciplinelor (Disciplina) susținute de studenți cu nota medie de promovare la examen mai mare de 7, in ordine descrescătoare după denumirea disciplinei.
23. Să se afișeze lista disciplinelor (Disciplina) predate de cel puțin doi profesori.
24. in ce grupe de studii (Cod\_ Grupa) figurează mai mult de 24 de studenți?
25. Găsiți numele, prenumele și adresele studenților și ale profesorilor care locuiesc pe strada 31 August.
26. Afișați studenții (identificatorii) care au susținut (evaluare examen) la toate disciplinele predate de prof. lon.
27. Găsiți numele, prenumele și media grupei studenților care au susținut toate disciplinele predate de profesorii ce locuiesc pe strada 31 August.
28. Cați studenți au studiat, deja, disciplina Baze de date in 2018 și 2019 și care este media lor la reușita curenta?
29. Fumizați numele și prenumele studenților, care au avut mai mult de doua dezaprobări într-un an la o disciplina.
30. Fumizați numele, prenumele și media notelor pe grupe pentru studenți
31. Găsiți numele și prenumele studenților, care nu au luat nota. de promovare la reușita curenta la nicio disciplina.
32. Găsiți numele și prenumele studenților, care nu au luat note de promovare (inclusiv note intermediare) la nicio disciplina.
33. Găsiți denumirile disciplinelor și media notelor pe disciplina. Afișați numai disciplinele cu medii mai mari de 7.0.
34. Găsiți numele, prenumele și adresele studenților și ale profesorilor care locuiesc într-o localitate. Sa se afișeze denumirea localității și numărul de locuitori incluși 'in cele doua categorii. Datele se afișează 'in ordinea crescătoare a numarului membrilor din categoria menționata anterior.
35. Găsiți disciplina susținuta de studenți cu nota medie (la examen) cea mai înaltă.
36. Furnizați denumirile disciplinelor cu o medie mai mica decât media notelor de la disciplina Baze de date.
37. Găsiți denumirile disciplinelor la care nu au susținut examenul, in medie, peste 5% de studenți.

## **3.1 Descrierea etapelor realizate**

PostgreSQL este un sistem de gestionare a bazelor de date relaționale (RDBMS) dezvoltat de comunitatea open-source. Acesta utilizează ca limbaj principal de interogare SQL, împreună cu extensii specifice cum ar fi PL/pgSQL.

PostgreSQL oferă suport pentru cele mai recente versiuni ale limbajului SQL și oferă funcționalități avansate pentru gestionarea bazelor de date. Este o soluție robustă și scalabilă, potrivită pentru gestionarea unor volume mari de date și pentru aplicații de dimensiuni variate.

Unul dintre avantajele majore ale PostgreSQL este natura sa open-source, ceea ce înseamnă că este gratuit de utilizat și că comunitatea dezvoltatorilor contribuie constant la îmbunătățirea sa. Acest lucru asigură actualizări regulate și îmbunătățiri de securitate.

De asemenea, un alt beneficiu important este capacitatea PostgreSQL de a gestiona complexitatea operațiilor pe bază de date. Poate manipula seturi mari de date și poate executa operațiuni complexe de interogare și agregare cu eficiență.

În plus, PostgreSQL oferă facilități puternice de securitate, inclusiv autentificare avansată, gestionarea permisiunilor și criptarea datelor. Aceste caracteristici contribuie la asigurarea securității datelor stocate în baza de date.

În concluzie, PostgreSQL este o alegere excelentă pentru gestionarea bazelor de date relaționale, oferind o combinație de performanță, securitate și flexibilitate, totul într-un mediu open-source.

1.

select \*

from grupe;

2.

select \*

from discipline

order by nr\_ore\_plan\_disciplina desc;

3.

select distinct discipline.disciplina, profesori.nume\_profesor, profesori.prenume\_profesor

from studenti\_reusita

join discipline on studenti\_reusita.id\_disciplina = discipline.id\_disciplina

join profesori on studenti\_reusita.id\_profesor = profesori.id\_profesor

order by nume\_profesor, prenume\_profesor;

4.

select discipline.disciplina

from discipline

where len(discipline.disciplina) > 20;

5.

select \*

from studenti

where nume\_student like '%u';

6.

SELECT studenti.nume\_student, studenti.prenume\_student

FROM studenti

JOIN studenti\_reusita ON studenti.id\_student = studenti\_reusita.id\_student

WHERE studenti\_reusita.tip\_evaluare = 'Evaluarea2'

AND id\_disciplina = 30

ORDER BY nota DESC

LIMIT 5;

7.

select distinct cod\_grupa

from grupe

join studenti\_reusita on grupe.id\_grupa = studenti\_reusita.id\_grupa

join studenti on studenti.id\_student = studenti\_reusita.id\_student

where adresa\_postala\_student like '%31 August%';

8.

SELECT DISTINCT studenti.id\_student, nume\_student, prenume\_student

FROM studenti\_reusita

JOIN studenti ON studenti\_reusita.id\_student = studenti.id\_student

WHERE EXTRACT(YEAR FROM data\_evaluare) = 2016

AND tip\_evaluare = 'Examen';

9.

select distinct nume\_student, adresa\_postala\_student, id\_disciplina

from studenti

join studenti\_reusita on studenti\_reusita.id\_student = studenti.id\_student

where EXTRACT(YEAR FROM data\_evaluare) = 2018

and nota > 8;

10.

select nume\_student, prenume\_student

from studenti

join studenti\_reusita on studenti\_reusita.id\_student = studenti.id\_student

where id\_disciplina = 30

and tip\_evaluare = 'Examen'

and EXTRACT(YEAR FROM data\_evaluare) = 2018

and (nota < 8 and nota > 4);

11.

select distinct nume\_profesor, prenume\_profesor

from profesori

join studenti\_reusita on studenti\_reusita.id\_profesor = profesori.id\_profesor

where id\_disciplina = 30

and EXTRACT(YEAR FROM data\_evaluare) = 2018

and nota < 5

and tip\_evaluare = 'Examena';

12.

select distinct disciplina

from studenti\_reusita

join studenti on studenti.id\_student = studenti\_reusita.id\_student

join discipline on discipline.id\_disciplina = studenti\_reusita.id\_disciplina

where nume\_student = 'Popescu'

and prenume\_student = 'Ion';

13.

select nume\_student, prenume\_student, disciplina

from studenti\_reusita

join studenti on studenti.id\_student = studenti\_reusita.id\_student

join discipline on discipline.id\_disciplina = studenti\_reusita.id\_disciplina

where tip\_evaluare = 'Examen'

and nota > 8;

14.

select nume\_student, prenume\_student

from studenti\_reusita

join studenti on studenti.id\_student = studenti\_reusita.id\_student

join profesori on profesori.id\_profesor = studenti\_reusita.id\_profesor

where tip\_evaluare = 'Examen'

and prenume\_profesor = 'Ion'

and EXTRACT(YEAR FROM data\_evaluare) = 2017

union

select distinct nume\_student, prenume\_student

from studenti\_reusita

join studenti on studenti.id\_student = studenti\_reusita.id\_student

join profesori on profesori.id\_profesor = studenti\_reusita.id\_profesor

where tip\_evaluare = 'Examen'

and prenume\_profesor = 'George'

and EXTRACT(YEAR FROM data\_evaluare) = 2017;

15.

select nume\_student, prenume\_student, id\_profesor

from studenti\_reusita

join studenti s on studenti\_reusita.id\_student = s.id\_student

join discipline d on d.id\_disciplina = studenti\_reusita.id\_disciplina

where nr\_ore\_plan\_disciplina < 60

group by nume\_student, prenume\_student, id\_profesor;

16.

select nume\_profesor, prenume\_profesor

from studenti\_reusita

join profesori p on p.id\_profesor = studenti\_reusita.id\_profesor

where id\_student = 1

group by nume\_profesor, prenume\_profesor;

17.

select distinct discipline.disciplina, profesori.nume\_profesor, profesori.prenume\_profesor

from studenti\_reusita

join discipline on studenti\_reusita.id\_disciplina = discipline.id\_disciplina

join profesori on studenti\_reusita.id\_profesor = profesori.id\_profesor

order by nume\_profesor, prenume\_profesor;

18.

select p.nume\_profesor, p.prenume\_profesor

from studenti\_reusita

join studenti s on studenti\_reusita.id\_student = s.id\_student

join profesori p on p.id\_profesor = studenti\_reusita.id\_profesor

where nume\_student = 'Dumitrescu'

group by nume\_profesor, prenume\_profesor;

19.

select count(nume\_student)

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

where tip\_evaluare = 'Evaluarea2'

and id\_disciplina = 3

and EXTRACT(YEAR from data\_evaluare) = 2023;

20.

select nume\_student, prenume\_student, count(nota)

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

group by nume\_student, prenume\_student;

21.

select nume\_profesor, prenume\_profesor, count(c) as discipline from (

select nume\_profesor, prenume\_profesor, count(distinct id\_disciplina) as c

from studenti\_reusita

join profesori p on p.id\_profesor = studenti\_reusita.id\_profesor

group by nume\_profesor, prenume\_profesor, id\_disciplina) t

group by nume\_profesor, prenume\_profesor

order by nume\_profesor, prenume\_profesor;

22.

select disciplina

from studenti\_reusita

join discipline d on d.id\_disciplina = studenti\_reusita.id\_disciplina

where tip\_evaluare = 'Examen'

and nota > 7

group by disciplina

order by disciplina desc;

23.

select disciplina from (

select disciplina, count(distinct id\_profesor) as c

from studenti\_reusita

join discipline d on d.id\_disciplina = studenti\_reusita.id\_disciplina

group by disciplina, id\_profesor) t

group by disciplina

having count(c) >= 2;

24.

select cod\_grupa from (

select cod\_grupa, count(distinct id\_student) as c

from studenti\_reusita

join grupe g on g.id\_grupa = studenti\_reusita.id\_grupa

group by cod\_grupa, id\_student) tab

group by cod\_grupa

having count(c) > 10;

25.

select nume\_student as nume, prenume\_student as prenume, adresa\_postala\_student as adresa

from studenti

where adresa\_postala\_student like '%Stefan%'

group by nume\_student, prenume\_student, adresa\_postala\_student

union

select nume\_profesor as nume, prenume\_profesor as prenume, adresa\_postala\_profesor as adresa

from profesori

where adresa\_postala\_profesor like '%Stefan%'

group by nume\_profesor, prenume\_profesor, adresa\_postala\_profesor;

26.

select id\_student from (

select id\_student, count(disc) as c from(

select id\_student, count(id\_disciplina) as disc

from studenti\_reusita

join profesori p on p.id\_profesor = studenti\_reusita.id\_profesor

where prenume\_profesor = 'Ana'

and tip\_evaluare = 'Examen'

group by

prenume\_profesor, id\_student, id\_disciplina) t

group by id\_student) u

where c in (

select count(distinct id\_disciplina)

from studenti\_reusita

join profesori p2 on p2.id\_profesor = studenti\_reusita.id\_profesor

where prenume\_profesor = 'Ana'

);

27.

select nume\_student, prenume\_student

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

where tip\_evaluare = 'Examen'

and nota < (

select min(distinct nota)

from studenti\_reusita

where id\_student = 100)

group by nume\_student, prenume\_student;

28.

select nume\_student, prenume\_student, avg(nota) as av from (

select nume\_student, prenume\_student, nota

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

where s.id\_student in (

select id\_student

from studenti\_reusita

join discipline d on d.id\_disciplina = studenti\_reusita.id\_disciplina

where (data\_evaluare like '2018%')

and disciplina = 'baze de date'

and tip\_evaluare = 'reusita curenta'

group by id\_student

intersect

select id\_student

from studenti\_reusita

join discipline d on d.id\_disciplina = studenti\_reusita.id\_disciplina

where (data\_evaluare like '2019%')

and disciplina = 'baze de date'

and tip\_evaluare = 'reusita curenta'

group by id\_student

) and tip\_evaluare = 'reusita curenta') t

group by nume\_student, prenume\_student;

29.

select nume\_student, prenume\_student, count(nota) as note, count(distinct id\_disciplina) as discipline

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

where nota < 5

and Extract(year from data\_evaluare) = 2017

group by nume\_student, prenume\_student, id\_disciplina, nota

having count(nota) > 2

union

select nume\_student, prenume\_student, count(nota) as note, count(distinct id\_disciplina) as discipline

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

where nota < 5

and Extract(year from data\_evaluare) = 2018

group by nume\_student, prenume\_student, id\_disciplina, nota

having count(nota) > 2;

30.

select distinct nume\_student, prenume\_student, avg(cast(nota as float)) over (

partition by studenti\_reusita.id\_grupa

) as media\_grupei

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

group by nume\_student, prenume\_student, id\_grupa, nota

order by media\_grupei;

31.

select nume\_student, prenume\_student from

studenti

except

select distinct nume\_student, prenume\_student

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

where nota >= 5

and tip\_evaluare = 'Examen';

32.

select nume\_student, prenume\_student from

studenti

except

select distinct nume\_student, prenume\_student

from studenti\_reusita

join studenti s on s.id\_student = studenti\_reusita.id\_student

where nota < 5;

33.

select disciplina, average from (

select id\_disciplina, avg(cast(nota as float)) as average from (

select id\_disciplina, nota

from studenti\_reusita

group by id\_disciplina, nota, tip\_evaluare) t

group by id\_disciplina) u

join discipline on u.id\_disciplina = discipline.id\_disciplina

where average > 7.0;

34.

SELECT localitate, adresa, nume, prenume, COUNT(numar) OVER (PARTITION BY localitate) AS locuitori

FROM (

SELECT

substring(adresa, 1, position(',' in adresa)) as localitate,

nume,

prenume,

adresa,

COUNT(nume) as numar

FROM (

SELECT DISTINCT nume\_student as nume, prenume\_student as prenume, adresa\_postala\_student as adresa

FROM studenti

UNION

SELECT DISTINCT nume\_profesor as nume, prenume\_profesor as prenume, adresa\_postala\_profesor as adresa

FROM profesori

) t

GROUP BY adresa, prenume, nume, substring(adresa, 1, position(',' in adresa))

) g

GROUP BY localitate, adresa, nume, prenume, numar

ORDER BY locuitori, localitate;

35.

select disciplina from (

select id\_disciplina, avg(cast(nota as float)) as average from (

select id\_disciplina, nota

from studenti\_reusita

where tip\_evaluare = 'examen'

group by id\_disciplina, nota) t

group by id\_disciplina) u

join discipline on u.id\_disciplina = discipline.id\_disciplina

where average = (

select max(average) from (

select avg(cast(nota as float)) as average from (

select id\_disciplina, nota

from studenti\_reusita

where tip\_evaluare = 'examen'

group by id\_disciplina, nota) t

group by id\_disciplina) u);

36.

select disciplina, average from (

select id\_disciplina, avg(cast(nota as float)) as average from (

select id\_disciplina, nota

from studenti\_reusita

group by id\_disciplina, nota, tip\_evaluare) t

group by id\_disciplina) u

join discipline on u.id\_disciplina = discipline.id\_disciplina

where average > (

select avg(cast(nota as float)) from (

select nota as nota

from studenti\_reusita

where studenti\_reusita.id\_disciplina = 30

group by studenti\_reusita.id\_disciplina, nota, tip\_evaluare

) m

);

37.

select disciplina, count(id\_student) as restanta, total from(

select id\_student, id\_disciplina

from studenti\_reusita

where tip\_evaluare = 'Examen'

and nota < 5

group by id\_student, id\_disciplina

) t, (

select id\_disciplina, count(distinct id\_student) as total

from studenti\_reusita

group by id\_disciplina

) g, discipline d

where t.id\_disciplina = d.id\_disciplina

and g.id\_disciplina = d.id\_disciplina

group by t.id\_disciplina, disciplina, total

having (cast(count(id\_student) as float) / cast(total as float) \* 100 ) > 5

order by t.id\_disciplina;

# 4 Concluzie

În urma efectuării lucrării de laborator cu numărul 2, am avut ocazia să explorez bazele de date și să învăț cum să gestionez fluxul de date în aplicații, site-uri și alte tehnologii. Am utilizat aplicația PostgreSQL pentru a rezolva sarcina dată, prin crearea, redactarea și ștergerea tabelelor în baza de date existentă numită "universitatea".

Acest laborator mi-a oferit o introducere în utilizarea bazelor de date și în instrumentul de lucru PostgreSQL. Aceste cunoștințe acumulate îmi vor fi utile în cadrul orelor de curs, în practică și în alte obiective viitoare. De asemenea, aceste abilități vor fi de folos în activitățile mele zilnice, dar și într-un context profesional, în cadrul unei companii care se ocupă de programare sau gestionare a datelor.